НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

«На правах рукопису»
УДК 004.852

«До захисту допущено»
Завідувач кафедри
_____ О.Л. Тимощук
«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо – науковою програмою

зі спеціальності 124 Системний аналіз

на тему: «Навчання з підкріпленням для довгострокового планування»

Виконав:
студент ІІ курсу, групи КА – 92мп
Титаренко Андрій Миколайович _____

Керівник:
директор ІПСА,
д.ф-м.н. проф.
Касьянов П.О. _____

Рецензент:
професор кафедри інт. та диф. рівнянь,
КНУ ім. Тараса Шевченка, д.ф-м.н. проф.
Капустян О.В. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ
2020

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – другий (магістерський)

Спеціальність – 124 «Системний аналіз»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ _____
(підпис) (ініціали, прізвище)

«___»_____20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Титаренка Андрія Миколайовича

1. Тема дисертації: «Навчання з підкріпленням для довгострокового планування», науковий керівник дисертації Касьянов Павло Олегович, д.ф-м.н., професор, затверджені наказом по університету від «02» листопада 2020 р. № 3182-с.

2. Термін подання студентом: 13.12.20.

3. Об'єкт дослідження: задача довгострокового планування із використанням методів глибокого навчання з підкріпленням.

4. Предмет дослідження: Методи підвищення ефективності методів глибокого навчання з підкріпленням для задач з довгостроковим плануванням.

5. Перелік завдань, які потрібно розробити:

1. Здійснити огляд технічної літератури за темою роботи;
2. Дослідити актуальність обраної теми;

3. Ознайомитись із існуючими методами глибокого навчання з підкріпленням та проблеми довгострокового планування;
4. Здійснити порівняльний аналіз наявних методів, проаналізувати їх переваги та недоліки;
5. Запропонувати і реалізувати метод навчання з підкріпленням для вирішення задач довгострокового планування;
6. Провести аналіз результатів;
7. Провести аналіз ринкових можливостей запуску стартап проекту;
8. Розробити концептуальні висновки;
9. Підготувати ілюстративний матеріал;
10. Оформити пояснювальну записку.

6. Орієнтовний перелік ілюстративного матеріалу:

1. Постановка завдання дослідження;
2. Існуючі методи навчання з підкріпленням;
3. Графіки та таблиці, що демонструють порівняння якості роботи методів для вибраної задачі;
4. Структура програмного продукту;
5. Розрахункові таблиці з результатами.

7. Дата видачі завдання: 5 вересня 2019 року.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Отримання завдання на магістерську дисертацію	01.09.2020 – 06.09.2020	
2.	Огляд технічної літератури за темою	07.09.2020 – 13.09.2020	
3.	Концептуальний вступ дисертації. Формулювання об’єкта, предмета, цілі, завдань, новизни, практичної значущості результатів. Дослідження актуальності обраної теми	14.09.2020 – 20.09.2020	
4.	Перший розділ. Аналіз існуючих методів для навчання з підкріпленням	21.09.2020 – 27.09.2020	
5.	Другий розділ. Аналіз підходів до задач довгострокового планування.	28.09.2020 – 04.10.2020	
6.	Розробка початкового програмного забезпечення	05.10.2020 – 11.10.2020	
7.	Третій розділ. Покращення якості програмного забезпечення	12.10.2020 – 18.10.2020	
8.	Аналіз результатів	19.10.2020 – 25.10.2020	
9.	Проведення аналізу ринкових можливостей стартап – проекту	26.10.2020 – 01.11.2020	
10.	Підготовка ілюстративного матеріалу	02.11.2020 – 15.11.2020	
11.	Оформлення пояснюальної записки	16.11.2020 – 27.11.2020	

Студент _____ А.М. Титаренко

Науковий керівник дисертації _____ П.О. Касьянов

РЕФЕРАТ

Магістерська дисертація: 91 с., 25 рис., 25 табл., 1 додаток, 44 джерел.

Об’єкт дослідження - задача довгострокового планування із використанням методів глибокого навчання з підкріпленням.

Предмет дослідження - методи підвищення ефективності методів глибокого навчання з підкріпленням для задач з довгостроковим плануванням.

Методами досліджень виступають методи статистичного аналізу та аналізу методів навчання з підкріпленням.

Мета даної роботи полягає у дослідженні та вдосконаленні існуючих методів навчання з підкріпленням для вирішення проблем із довгостроковим плануванням.

Актуальність теми: останні найбільші успіхи штучного інтелекту показують, що все складнішим стає використання існуючих методів глибокого навчання з підкріпленням для дедалі складніших задач. Однією з проблем є проблема довгострокового планування. Вона виникає, коли для успішного рішення задачі потребується ієрархія навичок та планування на декількох рівнях абстракції. Таким чином, вирішення цієї проблеми, суттєво розширить діапазон можливостей навчання з підкріпленням.

Результати роботи: запропонована модифікація методу моделей з темпоральними різницями для автоматичної торгівлі цінними паперами. Спроектовано та реалізовано програмний комплекс для експериментування та аналізу результатів.

Новизна роботи: запропоновано модифікацію методу моделей за темпоральними різницями для автоматичної торгівлі цінними паперами.

ЗАДАЧА ДОВГОСТРОКОВОГО ПЛАНУВАННЯ, НАВЧАННЯ З ПІДКРІПЛЕННЯМ, ГЛИБОКЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ, ЦІННІ ПАПЕРИ, МАРКОВСЬКІ ПРОЦЕСИ ПРИЙНЯТТЯ РІШЕНЬ, ЦІЛЕ-ЗАЛЕЖНІ СТРАТЕГІЇ

ABSTRACT

Master’s thesis explanatory note: 91 p., 25 tables, 25 fig.,1 application, 44 references.

The object of research is the task of long-term planning using deep reinforcement learning methods.

The subject of the research is methods of increasing the efficiency of deep reinforcement learning methods for tasks with long-term planning.

Research methods are methods of statistical analysis and analysis of reinforcement learning methods.

The purpose of this work is to research and improve existing reinforcement learning methods to solve problems with long-term planning.

Relevance of the topic: the latest great successes of artificial intelligence show that it is becoming increasingly difficult to use existing methods of deep reinforcement learning for increasingly complex tasks. One of the issues is the problem of long-term planning. It occurs when a hierarchy of skills and planning at several levels of abstraction is required to successfully solve a problem. Thus, solving this problem will significantly expand the range of reinforcement learning capabilities.

Results of work: the modification of a method of temporal difference models for automatic stock trading is proposed. A software package for experimentation and analysis of results has been designed and implemented.

Novelty of work: modification of a method of temporal difference models for automatic stock trading is proposed.

LONG-TERM PLANNING, REINFORCEMENT LEARNING, DEEP LEARNING, FORECASTING, STOCK TRADING, MARKOV DECISION PROCESSES, GOAL-CONDITIONED POLICIES

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

МППР – Марковський процес прийняття рішень;

НОС – наближена оптимізація стратегій;

ГН – глибоке навчання;

ШНМ – штучна нейронна мережа.

ВСТУП

Вже майже століття перед людством стоять проблеми робототехніки та штучного інтелекту, в вирішенні яких наука і техніка поступово покращують свої результати. До недавніх пір задачі комп'ютерного зору, прийняття рішень та управління вирішувалися з використовуванням апріорних знань в моделях та алгоритмах. Так, для задач класифікації [1 - 3] використовувалися рукотворні ознаки зображення, а для задач навігації - методи відтворення тривимірних поверхонь заснованих на проективній геометрії [4]. Протягом останнього десятиріччя відбувся інтенсивний розвиток глибокого навчання, причиною якого став розвиток обчислювальної техніки. Незалежні від доменної області, методи глибокого навчання перевершили найкращі алгоритми та методи з багатьох предметних областей таких, як комп'ютерний зір (зокрема в задачах класифікації зображень [5 - 7], детекції об'єктів [8, 9], семантичної сегментації [10], бінарної сегментації [11], генерування зображень [12] та інших), розпізнавання та генерування мови [13, 14], задач обробки природної мови. Крім того, як універсальні нелінійні апроксиматори функцій [15] нейронні мережі застосовуються і в задачах навчання з підкріпленням (керування, оптимізації, оптимізації недиференційованих функцій та інших). Такі моделі наразі успішно використовуються для багатьох ігор (нарди [16], шахи, шахи Го, Шогі [17, 18], Dota, StarCraft2), управління ресурсами та робототехніки. Останній напрямок зокрема виділяє багато підзадач керування роботами як на мікро- так і на макрорівні.

Об'єктом досліджень в даній роботі є задача довгострокового планування. В її рамках проводиться аналіз та порівняння існуючих методів навчання з підкріпленням для довгострокового планування. Пропонується метод підтримки рішень для торгівлі цінними паперами на базі Моделей із Темпоральними Різницями [19], що показує кращі результати ніж аналогічні підходи на базі глибокого навчання з підкріпленням. Надається аналіз отриманих стратегій.

В першому розділі розглядається задача довгострокового планування, а також визначені та описані Марковські процеси прийняття рішень з проміжними цілями, на термінологію яких досить природньо лягає задача навчання з підкріпленням для довгострокового планування.

В другому розділі надається огляд і детальний опис методів глибокого навчання з підкріпленням, їх класифікація, основні сильні сторони і недоліки, а також методи, що дозволяють дещо позбавитись останніх.

В третьому розділі розглядається практична задача автоматичної торгівлі цінними паперами. Надається опис існуючих підходів до рішення задачі за допомогою методів глибокого навчання з підкріпленням, та пропонується метод на базі Моделей із Темпоральними Різницями [19], [20-44].

В четвертому розділі надається опис набору даних по цінних паперах, розділ набору на підмножини для навчання та тестування. Описуються експерименти та моделі, що використовувалися в порівнянні. Аналізуються результати навчання моделей, отримані стратегії та їх успішність на модельних даних.

В п’ятому розділі пропонується та описується стартап проекту на базі запропонованих методів автоматичної торгівлі цінними паперами, надається аналіз ідеї, її життєздатність та можливі перспективи.

РОЗДІЛ 1 ЗАДАЧА ДОВГОСТРОКОВОГО ПЛАНУВАННЯ ТА ПІДХОДИ ДО ЇЇ ВИРІШЕННЯ

1.1 Опис задачі довгострокового планування

В задачах керування та прийняття рішень дуже важливу роль відіграє так званий горизонт планування. Він визначає наскільки довгострокові залежності можуть бути наявними в задачі. Для задач з великим горизонтом планування характерним є:

1. Необхідність врахування далекого минулого. Прикладами таких задач є задачі з неповною інформацією, де без довгострокової пам’яті дуже важко відрізнити одне спостереження від іншого. Така властивість характерна для задач візуальної навігації для роботів, торгових роботів, ігор з неповною інформацією (гра в «покер», «дурень») та багатьох інших.

2. Необхідність урахування довгострокового майбутнього. Така властивість характерна для дуже багатьох складних задач керування та прийняття рішень. Починаючи з настільних ігор, таких як Го, Шахи, Шогі, та завершаючи системами підтримки управління бізнесом. Насправді, цей аспект є набагато складнішим для вирішення, адже він включає в себе задачу прогнозування стану та дій всіх елементів системи (наприклад, інших учасників) та планування з урахуваням цього прогнозу.

3. Майбутнє із зворотним зв’язком. Наприклад, щоби прийняти рішення щодо наступного ходу в шахах або Го (рис 1.1), вам потрібно передбачити дії суперника, які в тому числі залежать від ваших дій у майбутньому (рис 1.2).

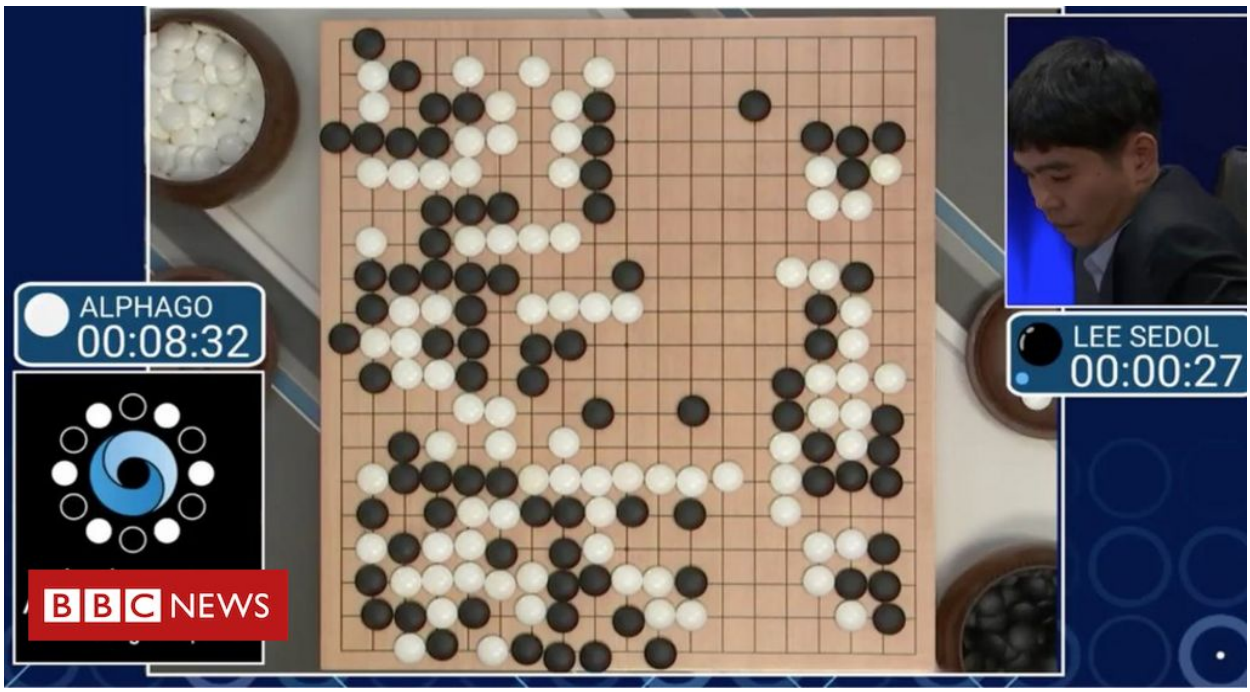

Рисунок 1.1 – Позиція на дошці в Го. Зображення взяте з відеозапису гри Лі Седоля проти AlphaGo, DeepMind Technologies.

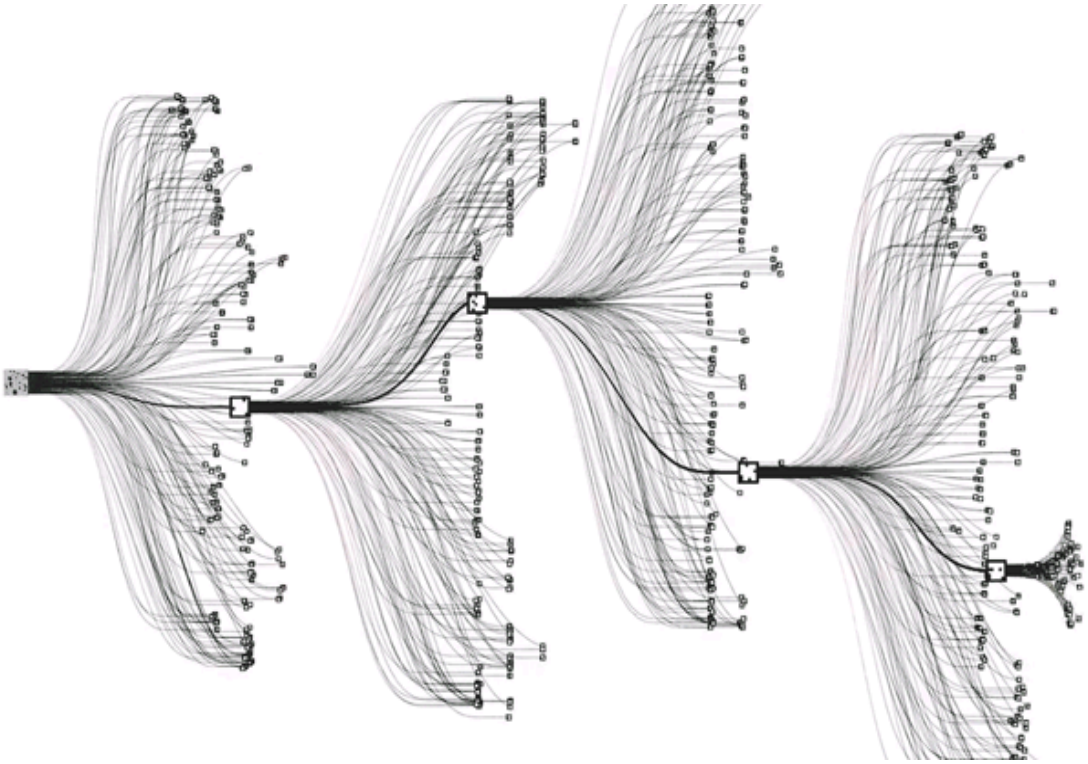

Рисунок 1.2 – Розгалуження дерева станів при довгостроковому плануванні. Зображення належить DeepMind Technologies.

1.2 Актуальність задач із довгостроковим плануванням

Задачі із довгостроковим плануванням наразі є дуже важливими для розвитку штучного інтелекту. Сучасні підходи частіше за все ж таки приймають досить локальні рішення, тобто імовірність отримання розумної стратегії, що, наприклад, на початку керування виконує дії, що дадуть ефект тільки наприкінці, використовуючи який можливо досягти набагато кращих результатів в задачі, є досить невеликою.

Однак, було б неправильно сказати, що успішних підходів таки не існує. В деяких випадках, можна використати апріорні знання про систему та робити багатокроковий прогноз. До таких випадків можна віднести настільні ігри, такі як Го та Шахи. Методи, що використовують нейронні мережі разом із Монте-Карло пошуком по дереву, показують дуже високі успіхи, перевершуючи найкращих гравців серед людей [10].

Проте, попри надлюдські результати в Го, дещо складніші в інших аспектах задачі вже не досягають таких чудових показників досить швидко. Проблеми бувають дуже різними. Починаючи з практичної неможливості належного пошуку стратегії, потреби в дуже великих обчислювальних можливостях закінчуючи надто складною структурою спостереження (наприклад, візуальне зображення) що суттєво ускладнює прогнозування.

Сучасні дослідження в області штучного інтелекту розглядають всі ці проблеми, і в деяких ситуаціях їх можна вирішити. Однак, в загальному випадку, задачі із довгостроковим плануванням практично неможливо вирішити без додаткових трюків, евристик, апріорних знань про систему і так далі.

1.3 Задача довгострокового планування як Марковський процес прийняття рішень

1.3.1 Відомості про Марковські процеси прийняття рішень

Марковським процесом прийняття рішень (МППР) називають набір з 4 множин (S, A, T, R), де:

1. $S -$ простір станів.
2. $A -$ простір дій.
3. $P_a(s, s') = \Pr(s_{t+1} = s' | s_t = s, a_t = a)$ – перехідні імовірності, за умови дії. $P_a(s, s')$ складають елементи оператору переходів T.

Нехай:

$$\mu_{t,j} = p(s_t = j)$$
$$\xi_{t,k} = p(a_t = k)$$
$$T_{i,j,k} = p(s_{t+1} = i | s_t = j, a_t = k)$$

Тоді має місце наступна рівність:

$$\mu_{t,i} = \sum_{j,k} T_{i,j,k} \mu_{t,j} \xi_{t,k},$$

4. $R(s, a)$ функція винагороди. Спів ставляє парі стану, в якому нині знаходиться агент, та дії, що він вибрав, скалярну величину – підкріплення. Задачею агенту є максимізація цієї функції.

Марковському процесу притаманна характерна марковська властивість, як те що, стан в є достатньою статистикою, тобто що при знанні теперішнього стану, не потрібно знати минулі стани для повного розуміння стану системи (рис. 1.3).

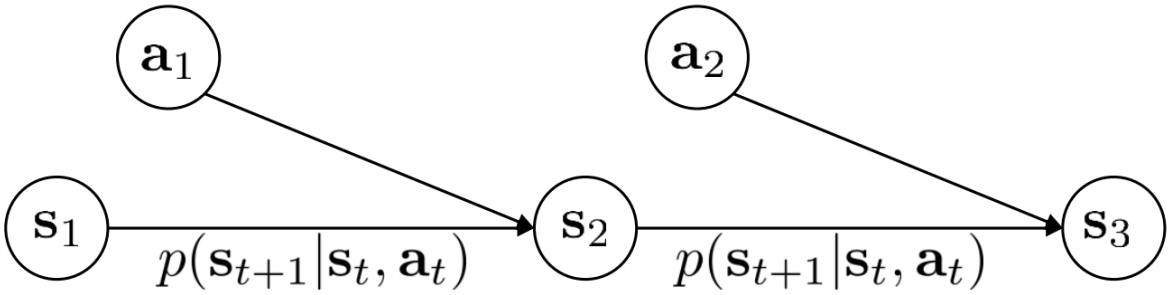

Рисунок 1.3 - Графічна модель МППР. Зображення взяте з лекцій курсу CS234 Deep Reinforcement Learning (UC Berkeley).

Крім того, існує розширення МППР – МППР із неповною інформацією (partially observed). Неповнота інформації означає, що функція стратегії приймає рішення тільки на основі спостережнь, а не повної інформації про нинішній стан s_t. Спостереження зазвичай позначають як o_t. Для прикладу такого МППР наведемо наступне середовище - лабіринт, де агент спостерігає тільки деякий окіл навколо його позиції (рисунок 1.4).

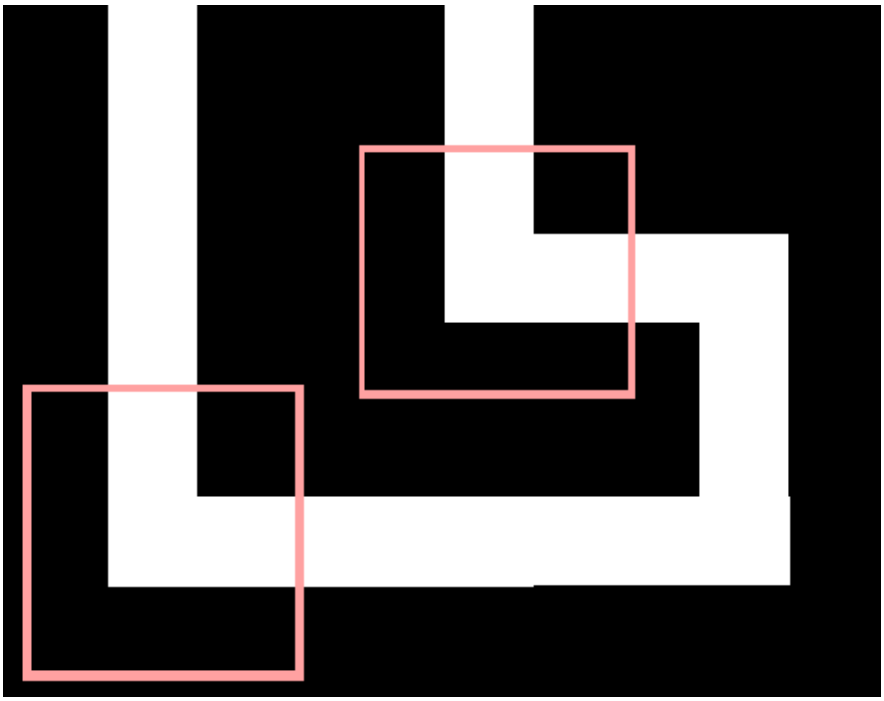

Рисунок 1.4 - МППР з неповними спостереженнями. На діаграмі помічені околи є спостереженнями, що агент бачить коли знаходиться в їх центрах.

Для МППР з неповною інформацією спостереження не визначає однозначно стан системи, тобто декільком станам може відповідати однакове спостереження, як на рисунку 1.4 Це одна з проблем, що дуже ускладнюють рішення таких МППР за допомогою звичайних алгоритмів.

Формально, (рисунок 1.5), такий МППР представляє набір із $(S, A, O, T, \mathrm{E}, R)$:

1. S – простір станів.
2. A – простір дій.
3. O – простір спостережень, що агент отримує від середовища в якості вхідної інформації.
4. T – оператор динаміки середовища, визначення якого співпадає із аналогом із визначення МППР.
5. E – тензор ймовірностей емісії спостережень $p(o_t|s_t)$, де $o_t \in O$.
6. R – функція винагороди, $R: S \times A \to \mathbb{R}$.

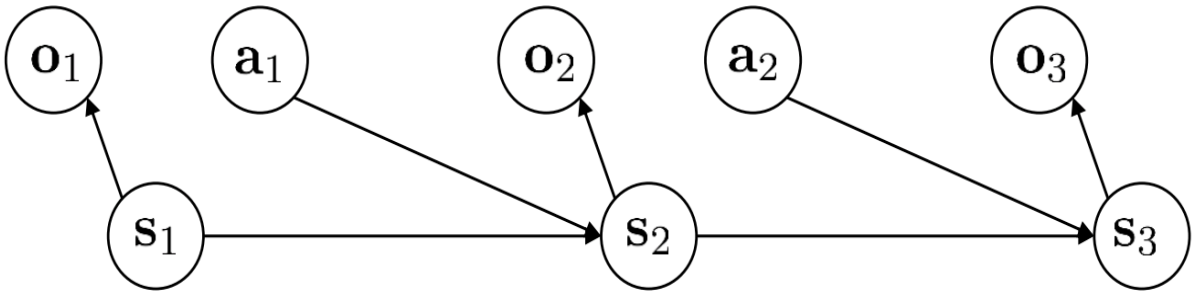

Рисунок 1.5 - МППР із неповною інформацією у вигляді графічної моделі. Зображення взяте з лекцій курсу CS234 Deep Reinforcement Learning (UC Berkeley)

1.3.2 Поняття стратегії

Функція стратегії (policy function) – це функція π, що за відображає простір станів S на множину розподілів $p(a \mid s)$ – розподілів дій які доступні агенту.

Функція стратегії називається детермінованою, коли співставлений розподіл є точковим, тобто його зображення складається із 1 та 0. Функція стратегії та ймовірностний оператор переходу однозначно визначають розподіл послідовності станів та дій $\tau = (s_1, a_1, \dots, s_T, a_T)$ - траєкторій.

$$p_\pi(\tau) = p(s_1) \prod_{t=1}^{T} \pi \quad (a_t|s_t) p(s_{t+1}|s_t, a_t)$$

В якості задачі в МППР алгоритм намагається отримати функцію стратегії $\pi^*(s)$, (у випадку МППР з неповною інформацією - $\pi^*(o)$), що б максимізувала математичне очікування кумулятивної винагороди:

$$\pi^* = argmax_\pi \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_t \gamma^t R(s_t, a_t) \right]$$

Даний процес можна проілюструвати наступною графічною моделлю (рис. 1.6).

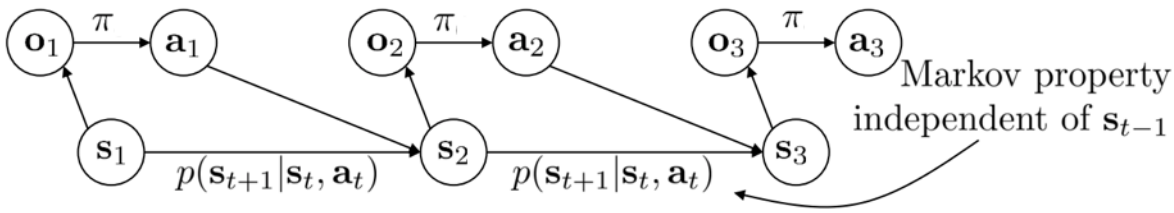

Рисунок 1.6 - МППР у вигляді графічної моделі. Зображення взяте з лекцій курсу CS234 Deep Reinforcement Learning (UC Berkeley)

Пошук оптимальної функції стратегії має сенс, адже для випадку Марковських процесів прийняття рішень із скінченними множинами дій та множинами станів існує теорема, що гарантує існування оптимальної

марковської інформаційної стратегії, яка максимізує математичне очікування кумулятивної винагороди [49].

1.3.3 Особливості МППР для довгострокового планування

Розглянемо наступні особливості МППР. По-перше, великий коефіцієнт дисконтування призводить до проблем із дисперсією дисконтованої суми винагород. Також, дуже велика кількість алгоритмів навчання з підкріпленням характеризується теоретичними оцінками помилки та збіжності, які часто є невизначеними для $\gamma \to 1$.

По-друге, в цих задачах, передбачається, що оптимальна стратегія повинна ураховувати довгострокове минуле, тобто спостереження не несуть багато інформації і дуже відрізняються від стану.

По-третє, для довгострокового планування, потрібно явно чи неявно оцінити $p(s_{t+1}|s_t, a_t)$ – модель середовища. Ця модель може включати дії інших агентів, а крім того, ускладнює планування багатьох дій вперед. Методи пошуку по дереву Монте-Карло, наприклад, використовують апріорне знання моделі, для багаторазового програвання епізоду та усереднення винагород. Проте, в цій роботі ми не будемо розглядати такі випадки.

В загальному випадку модель середовища не є відомою, а також є досить важко оцінюємою для складних видів спостережень. Також варто зазначити, що $p(s_{t+1}|s_t, a_t)$ включає в себе s_t як предмет прогнозу, проте у випадку МППР із неповною інформацією, насправді частіше за все ми оцінюємо або $p(o_{t+1}|o_t, a_t)$ або $p(z_{t+1}|z_t, a_t)$, де z_t – прихований стан моделі середовища, що визначається під час навчання.

1.3.4 МППР із проміжними цілями

В даній роботі, ми розглядаємо розширення Марковського процесу прийняття рішень - МППР з проміжними цілями.

МППР з проміжними цілями визначається кортежем:

$$(S, G, A, p, R, T_{max}, \rho_0, \rho_g),$$

де S – простір станів,

G – множина цілей,

A – простір дій,

$p(s_{t+1}|s_t, a_t)$ – функція динаміки інваріантна до часу,

R – функція винагороди,

T_{max} – максимальний горизонт,

ρ_0 – розподіл початкових станів,

ρ_g – розподіл цілей.

Основною задачею навчання з підкріпленням з проміжними цілями є отримання функції стратегії $\pi(a_t|s_t, g, t)$ що максимізує очікувану суму винагород $\mathbb{E}[\sum_{t=0}^{T_{max}} R(s_t, g, t)]$, де ціль g береться випадково з розподілом ρ_g і станами розподіленими відповідно до $s_0 \sim \rho_0$, $a_t \sim \pi(a_t|s_t, g, t)$, $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$. В даній роботі розглядається випадок, в якому множина цілей співпадає із простором станів, тобто $G = S$.

Важливою компонентою МППР із проміжними цілями є функція вартості за умовою цілі V^π, що прогнозує очікувану суму майбутніх винагород, за умови поточного стану s, цілі g в момент часу t:

$$V^{\pi}(s,g,t) = \mathbb{E}\left[\sum_{t'=t}^{T_{max}} R(s_{t'},g,t') \mid s_t = s, \pi \text{ за умови } g\right]$$

Попри те, що різноманітні функції вартості можуть бути використаними, в цій роботі ми скористаємося наступною:

$$R_{TDM}(s,g,t) = -\delta(t = T_{max})d(s,g),$$

де δ – індикаторна функція, а функція відстані d визнається залежно від задачі.

Такий вибір функції винагороди дає наступну інтерпретацію. При відомому стані s, як близько до цілі g агент, що використовує стратегію π за умови цілі g, може дістатися за t часових кроків?

Функцію вартості в такому випадку можна використовувати для оцінювання міри досяжності певного стану від заданого стану за час t. Таким чином, ми отримуємо можливість роботи із темпоральними абстракціями.

Проте, варто зазначити, що такий підхід працює добре тільки за умови того, що ціль g відповідає насправді існуючому стану, тобто $g \in S$. Це важливе обмеження змушує нас використовувати представлення станів, звужуючи пошук до існуючих станів. У випадку, коли простори станів є багато-вимірними (>~100), це може суттєво сповільнити дослідження середовища.

В наступних розділах, надається опис методів та модифікації, що створені, щоби вирішити цю проблему.

1.4 Висновки

В цьому розділі була розглянута задача довгострокового планування та її актуальність, а також оглянутий концепт Марковських процесів прийняття рішень. Описано і класифіковано задачі довгострокового планування, наведені аргументи, що оправдують необхідність її дослідження для застосувань в різних сферах людської діяльності.

В рамках опису поняття МППР було надане їх визначення, деякі властивості. Були розлянуті МППР з неповною інформацією, наведені рівняння, що характеризують динаміку середовища та взаємодії середовища з агентом в рамках цих процесів. Було визначено поняття функції стратегії та головної задачі агенту діючого в МППР. Була переформульована задача довгострокового планування в термінах МППР з проміжними цілями, що дає можливість використовувати підходи до їх вирішення для задач долгострокового планування. Відповідні методи були оглянуті в другому розділі.

В кінці розділу була дана характеристика складності задачі довгострокового планування, основні її джерела та проблеми, ними породжені. Крім того, був наданий опис Марковських процесів прийняття рішень із проміжними цілями [19], в рамках яких можна переформулювати задачу довгострокового планування для більш ефективного вирішення цієї проблеми.

РОЗДІЛ 2 МЕТОДИ ГЛИБОКОГО НАВЧАННЯ З ПІДКРІПЛЕННЯМ

2.1 Основні поняття навчання з підкріпленням

2.1.1 Стратегії та апроксимація функцій стратегії

Функції стратегії можна умовно поділити на детерміновані та випадкові. Детермінованою функцією стратегією є невипадкова функція:

$$\mu: o_t \longmapsto a_t,$$

що за спостереженням $o_t \in O$ визначає наступну дію агенту $a_t \in A$.

Також зауважимо, що МППР втрачає марківську властивість, цю функцію стратегії можна змінити наступним чином для врахування інформації про довшу історію станів, для хоча б часткового переведення такого процесу прийняття рішень на МППР. Таке переведення можна реалізувати, через введення деякого внутрішнього стану функції h_t:

$$\mu: (o_t, h_{t-1}) \longmapsto (a_t, h_t)$$

Випадковою, або стохастичною, функцію стратегії називають стохастичну функцію:

$$a_t \sim \pi(\cdot\,|s_t)$$

На практиці, це реалізується як функція, що видає розподіл:

$$\pi(a_t|s_t) = p(a_t|s_t)$$

Тобто відображаючи простір станів, або спостережень МППР на простір розподілів на просторі дій, з якого генерується дія. Для неперервних просторів

дій, розподіл часто моделюють як діагональний гаусівський, з очікуванням в $\pi(a_t|s_t)$, та діагональною матрицею коваріації, елементи якої в найпростішому випадку, апроксимується окремим чином.

Функцію стратегії можна представляти по-різному. В деяких алгоритмах навчання з підкріпленням, де функція стратегії не наближається, її представляють табличним методом. Проте, через експоненційий ріст кількості елементів, при збільшенні розмірності множини можливих дій, або множини станів, цей метод втрачає ефективність через неможливість зберігання таблиці в пам'яті, алгоритмічну складність операцій над цією таблицею.

В рамках глибокого навчання з підкріпленням, функцію стратегії апроксимують нелінійними параметричними функціями - нейронними мережами. Функцію позначають через π_θ, або μ_θ, де θ – вектор усіх параметрів глибокої параметричної моделі. Задачу в даному випадку, можна сформулювати так:

$$p_\pi(\tau) = p(s_1)\prod_{t=1}^{T}\pi_\theta(a_t|s_t)p(s_{t+1}|s_t,a_t),$$

$$\theta^* = argmax_{\pi_\theta}\mathbb{E}_{\tau\sim p_\theta(\tau)}\left[\sum_t R(s_t,a_t)\right]$$

2.1.2 Кумулятивна винагорода та поняття функції вартості.

Кумулятивна винагорода в МППР із скінченним горизонтом та без коефіцієнту дисконтування (finite-horizon undiscounted return) визначається наступним чином:

$$R(\tau) = \sum_{t=0}^{T} R(s_t, a_t)$$

Прикладом такого МППР, де така винагорода має місце є середовище, де стратегії контролюють штучною роботичною рукою з ціллю виконання завдання, що включає маніпулювання тканиною за обмежену кількість кроків (рисунок 2.2).

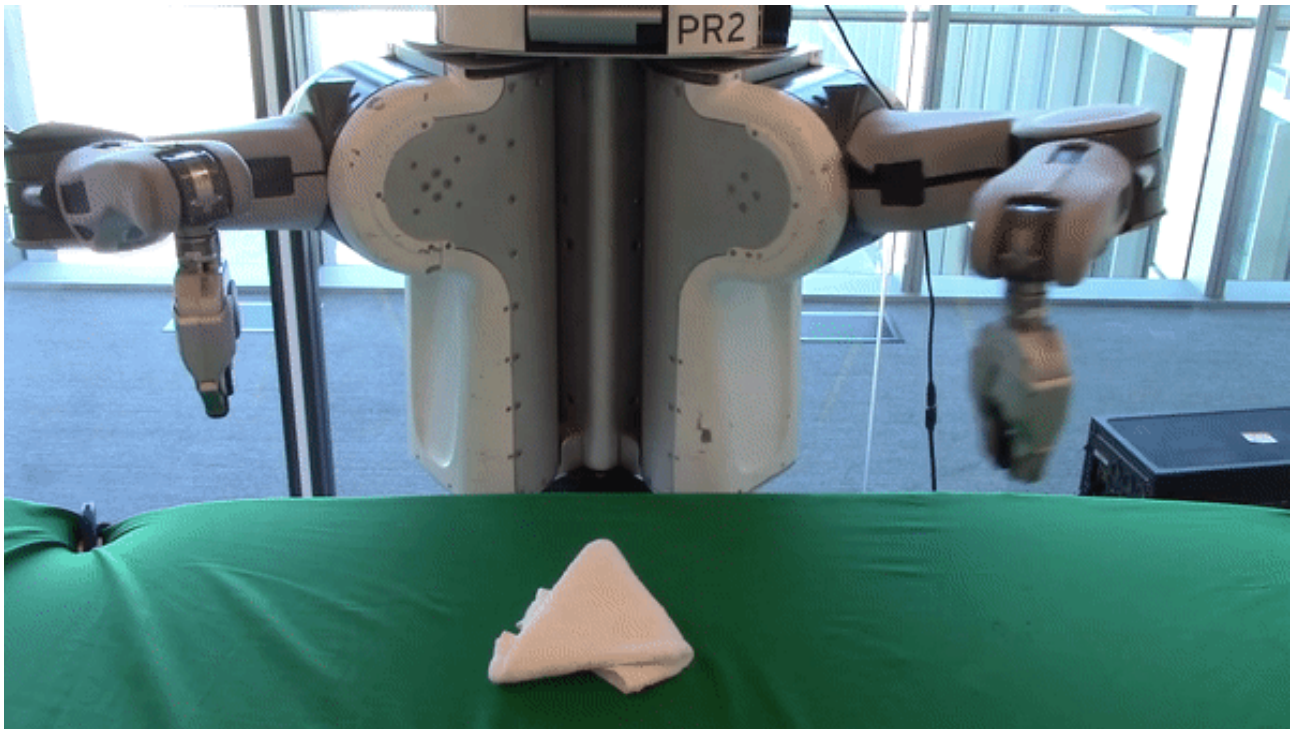

Рисунок 2.2 – Роботична рука згортає шмат такнини. Зображення взяте із блогу Berkeley AI Research.

Кумулятивна винагорода в МППР із нескінченним горизонтом із визначеним коефіцієнтом дисконтування визначається наступним чином:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t),$$

де γ – коефіцієнт МППР дисконтування, $0 < \gamma < 1$.

Використання коефіцієнту дисконтування має декілька переваг. По-перше, отримуємо гарантію збіжності ряду при умові обмеженості функції винагороди. По-друге, його використання приймає наступну інтерпретацію. Агент цінує більше винагороду, що прийде раніше, тобто надає переваги нагороді зараз, ніж потім. Такий механізм узгоджується з теорією цінностей. По-третє, цей коефіцієнт можна інтерпретувати, як ймовірність переходу в деякий новий стан МППР – воронку, іншими словами – імовірність смерті агенту в кожен момент часу.

Функція вартості - функція $V^{\pi}(s)$, що відображає простір станів на множину очікуваних кумулятивних винагород для кожного стану із простору за умови того, що цей стан – початковий.

$$V^{\pi}(s) = \mathbb{E}_{\tau\sim\pi}[R(\tau)|s_0 = s]$$

Q-функцією - функція $Q^{\pi}(s,a)$, що відображає простір станів на множину очікуваних кумулятивних винагород для кожного стану із простору та дії із простору за умови того, що цей стан – початковий, а дія із простору - перша.

$$Q^{\pi}(s,a) = \mathbb{E}_{\tau\sim\pi}[R(\tau)|s_0 = s, a_0 = a]$$

Оптимальна функція вартості $V^{*}(s)$ - функція вартості, що генерується, якщо вибрана стратегія - оптимальна π^{*}:

$$V^{*}(s) = \max_{\pi}(\mathbb{E}_{\tau\sim\pi}[R(\tau)|s_0 = s])$$

Оптимальною Q-функція $Q^{*}(s,a)$ - Q-функція, що генерується, якщо вибрана стратегія – оптимальна π^{*}:

$$Q^{*}(s,a) = \max_{\pi}(\mathbb{E}_{\tau\sim\pi}[R(\tau)|s_0 = s, a_0 = a])$$

Можна довести наступний зв’язок між цими функціями:

$$V^{\pi}(s) = \mathbb{E}_{a\sim\pi}[Q^{\pi}(s,a)],$$
$$V^{*}(s) = \max_{\text{a}}[Q^{*}(s,a)]$$

Для функцій вартості можна сформулювати наступну залежність - Рівняння Белмана:

$$V^{\pi}(s) = \mathbb{E}_{a\sim\pi(\cdot|S),s'\sim P(\cdot|S,a)}[R(s,a) + \gamma V^{\pi}(s')],$$
$$Q^{\pi}(s,a) = \mathbb{E}_{s'\sim P(\cdot|S,a)}\big[R(s,a) + \gamma\mathbb{E}_{a'\sim\pi(\cdot|S)}[Q^{\pi}(s',a')]\big],$$

де $P(\cdot\,|s,a)$ –розподіл станів переходу за умовою стану s та дії a;

γ – коефіцієнт дисконтування.

Відповідно, для оптимальних функцій вартості Рівняння Белмана мають наступний вигляд:

$$V^{*}(s) = \max_{\text{a}}\mathbb{E}_{s'\sim P(\cdot|S,a)}[R(s,a) + \gamma V^{*}(s')],$$
$$Q^{*}(s,a) = \mathbb{E}_{s'\sim P(\cdot|S,a)}\big[R(s,a) + \gamma\max\nolimits_{a'}[Q^{*}(s',a')]\big]$$

На основі рівняння Белмана можна сформулювати підходи до оптимізації функції стратегії, що випливає із функцій вартості. Два класичних загальних підходи, що є фундаментальними і досить абстрактними є алгоритми Ітерації Вартостями й Ітерації Стратегіями.

Перший - Ітерація Вартостями – визначає стан, що оптимізує функцію вартостей. Таким чином, ітеративним чином покращуємо функцію $V(s)$ та, відповідно, стратегію. Ініціалізуємо функцію вартостей $V(s)$ випадковим представленням. Далі послідовно оновлюються представлення функцій $Q(s,a)$ та $V(s)$ до збіжності. У випадку звичайних МППР, збіжність алгоритму

гарантована. Детальна схема цього алгоритму зображена на наступному рисунку 2.3.

Повторювати:

Для всіх $s \in S$

Для всіх $a \in A$

$$Q(s,a) \leftarrow \mathbb{E}[R|s,a] + \gamma \sum_{s' \in S} P(s'|s,a)V(s')$$

$$V(s) \leftarrow \max_a Q(s,a)$$

До збіжності $V(s)$

Рисунок 2.3 - Алгоритму Ітерації Вартостями

Другий алгоритм - Ітерація Стратегіями – представляє собою ітерацію представленнями стратегії до збіжності. Детальна схема цього алгоритму зображена на наступному рисунку 2.4.

Ініціалізувати план π' довільним чином

Повторювати:

$$\pi \leftarrow \pi'$$

Обчислити вартості за допомогою π вирішуючи СЛАР:

$$V^{\pi}(s) = \mathbb{E}[R|s,\pi(s)] + \gamma \sum_{s' \in S} P\big(s'\big|s,\pi(s)\big)V^{\pi}(s')$$

Покращуємо план для кожного стану:

$$\pi'(s) \leftarrow argmax_a(\mathbb{E}[R|s,a] + \gamma \sum_{s' \in S} P(s'|s,a)V^{\pi}(s'))$$

До збіжності π $(\pi = \pi')$

Рисунок 2.4 - Алгоритму Ітерації Стратегіями

Збіжність даних алгоритмів гарантована в звичайних МППР та деяких його модифікаціях [22]. Крім того, зауважимо, що представлення повинно бути табличним або, при деяких умовах, лінійним для збіжності. Проблемою цих алгоритмів є те, що для їх роботи треба знати динаміку середовища, тобто імовірнісний оператор переходів.

2.2 Класифікація алгоритмів навчання з підкріпленням

На наступній схемі зображена структура сімейств алгоритмів із прикладами [23 - 32] (рисунок 2.5):

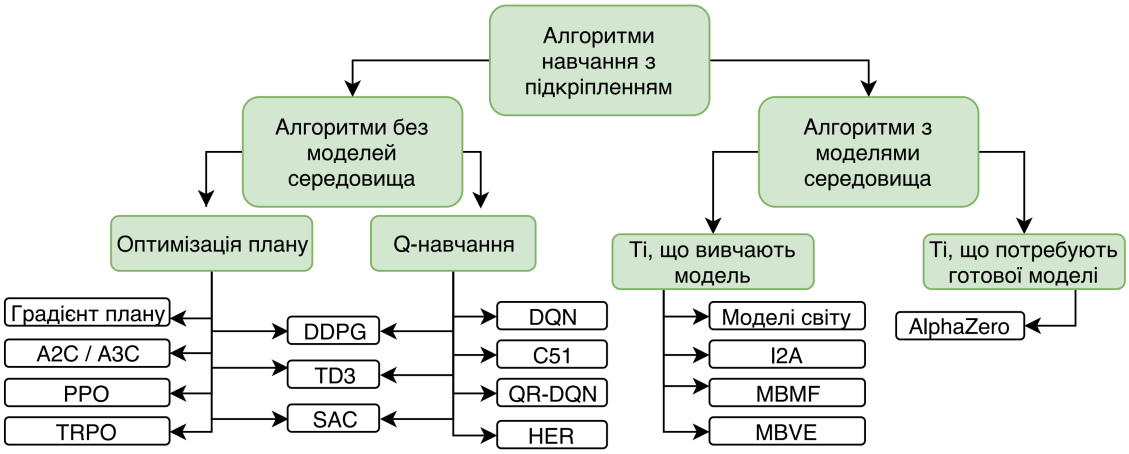

Рисунок 2.5 - Алгоритми глибокого навчання з підкріпленням

Загалом, алгоритми мають наступну структуру (рис 2.6).

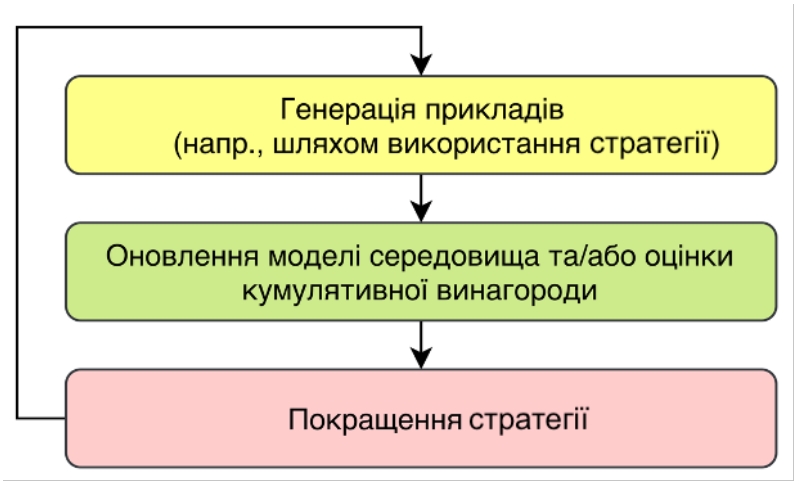

Рисунок 2.6 - Схема типового алгоритму глибокого навчання з підкріпленням

Типовий алгоритм можна представити як повторення трьох послідовних етапів.

Перший крок – отримання нових записів. Тут, запис - це кортеж:

$$(s_t, R_t, a_t, s_{t+1}),$$

де s_t – нинішній стан чи спостереження наступного стану,

a_t – застосована дія,

R_t – винагорода, що була отримана після застосування дії,

s_{t+1} – наступний стан або спостереження середовища.

Другий крок – застосування деякого оновлення параметрів додаткових апроксиматорів, що використовуються в алгоритмі.

Останній, третій крок – оновлення і покращення параметрів функції стратегії.

Зауважимо, що не всі кроки обов’язково присутні у всіх алгоритмах. Крім того, не завжди алгоритми явно наближують функцію стратегії, а наприклад, апроксимують функцію вартості залежну від дії.

2.2.1 Навчання з підкріпленням із моделями

Як було зазначено в попередньому розділі, алгоритми глибокого навчання з підкріпленням діляться на два види за наявністю явної апроксимації динаміки середовища. В цьому підрозділі розглядаються ті, що використовують моделі середовищ.

Ці алгоритми також можна умовно поділити на ті, в яких наближення функції динаміки середовища відбувається під час загального навчання, а також ті, де модель середовища відома, або передбачається користувачем.

До першого типу відносять, наприклад Моделі Світу, Агенти, що використовують уявлення [33], Навчання з підкріпленням із моделями із безмодельним уточненням [34].

До другого ж типу, відносять відомий метод від DeepMind AlphaZero [18] та AlphaGo [17], який відомий тим, що переміг чемпіона з Го – Лі Седоля. Ці методи використовують Монте-Карло пошук по дереву. Вхідне представлення складалося із спеціальних індексів на показників, а в AlphaZero – только стан ігрового поля та декілька модифікацій. Відсутність необхідності у спеціально підібраних ознаках в якості вхідного сигнала до алгоритму, дозволяє універсально використовувати даний алгоритм в середовищах, структура яких добре моделюється деревом. Таким чином, його використали в шахах, японських шахах Шогі та багатьох інших, отримуючи агенти, що перевершують майстерність кращих гравців.

2.2.2 Безмодельне навчання з підкріпленням

Інший тип алгоритмів глибокого навчання з підкріпленням – той, де динаміка середовища явним чином не наближається. Ці алгоритми ігнорують динаміку та на пряму оновлюють представлення функції стратегії ітеративним чином.

Такі алгоритми, в свою чергу, можна умовно розділити на два види – ті, що напряму оцінюють функцію стратегії, та ті, що напряму оцінюють функцію вартості, або Q-функція.

Оцінювання останньої лежить в основі алгоритмів Q-навчання. Такі методи характеризуються наступними властивостями:

1. За значенням Q-функції можна відтворити детерміновану функцію стратегії наступним чином:

$$\pi'(a_t|s_t) = \begin{cases} 1, \text{якщо } a_t = argmax_{a_t} Q^{\pi}(s_t, a_t) \\ 0, \text{в інших випадках} \end{cases}$$

2. Q-навчання зазвичай виконує мінімізацію функціоналу помилки Белмана, що напряму слідує із рівняння, і має наступний вигляд:

$$\mathcal{E} = \frac{1}{2}\mathbb{E}_{(s,a)\sim\beta}\left[Q_\theta(s,a) - \left[R(s,a) + \gamma \max_{a'} Q_\theta(s',a')\right]\right],$$

де β – загальний апріорний розподіл записів;

θ – вектор параметрів апроксиматора Q-функції.

Коли $\mathcal{E} = 0$, $Q_\theta(s,a) = R(s,a) + \gamma \max_{a'} Q_\theta(s',a')$, тобто в даній точці наближена Q функція є оптимальною. Даний алгоритм представлений на схемі 2.7.

Повторювати:
Генерація прикладів виду (s_i, a_i, s_{i+1}, r_i)
Повторити K разів:
$y_i \leftarrow R(s_i, a_i) + \gamma \max_{a_i'} Q_\theta(s_i', a_i')$
$\theta \leftarrow argmin_\theta \frac{1}{2}\sum_i \|Q_\theta(s_i, a_i) - y_i\|^2$
До збіжності

Рисунок 2.7 - Алгоритм Q-навчання.

У випадку глибокого навчання з підкріпленням, параметричні апроксиматори зазвичай є нелінійними (часто кусочно-лінійними, проте не лінійними). Тому, збіжність цієї процедури, на відміну від табличного аналогу не є гарантованою. Більш того, існують приклади розбіжності, а тому додаткові міри повинні прийматися для підвищення стабільності алгоритму.

Формальною причиною розбіжності є факт того, що оператор Белмана являє собою стискання Банаха в просторі із sup-нормою, а оператор апроксимації

(проектування на простір параметричних функцій) – за L2 нормою. Тому, результуючий оператор не є стиском, а тому збіжність довести не вийде.

Проте, це не означає, що алгоритм неможливо використовувати на практиці. Вони є досить відомими та використовуються в багатьох застосуваннях глибокого навчання з підкріпленням.

До алгоритмів із явною оптимізацією функції стратегії можна віднести методи із [23-25]. Про алгоритм із цього сімейства піде мова в наступному підрозділі.

Існують і методи, що лежать на межі між двома сімействами – [26, 28]. Один з таких алгоритмів також аналізується в цьому розділі.

2.3 Алгоритми Актор-Критик

2.3.1 Поняття переваги

Нагадаємо, що кумулятивною винагородою після застосування принципу причинності [37]:

$$\sum_{t\prime=t}^{T} R\left(s_{i,t\prime}, a_{i,t\prime}\right)$$

Цей вираз можна переписати у вигляді:

$$Q(s_t, a_t) = \sum_{t'=t}^{T} \mathbb{E}_{\pi_\theta}[R(s_{t'}, a_{t'})|s_t, a_t]$$

Базова вартість буде мати наступний вид:

$$V(s_t) = \mathbb{E}_{a_t \sim \pi_\theta(a_t, s_t)}[Q(s_t, a_t)]$$

Запишемо тепер градієнт параметризованої функції стратегії:

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left[\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta\left(a_{i,t}\middle|s_{i,t}\right) A^{\pi_\theta}\left(s_{i,t}, a_{i,t}\right)\right],$$

$$A^{\pi_\theta}\left(s_{i,t}, a_{i,t}\right) = Q\left(s_{i,t}, a_{i,t}\right) - V\left(s_{i,t}\right),$$

де A^{π_θ} - це так звана функція переваги.

На відміну від класичного градієнту стратегії, ця оцінка градієнту характеризується меншою дисперсією. Крім того, якість оцінювання функції переваги напряму впливає на дисперсія. Для оцінки переваги можна використати два підходи

Очевидно, що для цього, можна наблизити A^π, Q^π або V^π. Якщо маємо оцінку V^π, перевага приймає вигляд:

$$A^\pi(s_t, a_t) \approx R(s_t, a_t) + V^\pi(s_{t+1}) - V^\pi(s_t)$$

Функція переваги V^π може апроксимуватись додатковою нейромережею, або частина параметрів представлення функції стратегії можна використовуватися для зменшення складності. Для оцінки цієї функції вартості існує декілька видів оцінок:

1. Монте-Карло

$$V^\pi(s_t) \approx \sum_{t'=t}^{T} R(s_{t'}, a_{t'})$$

Характеризується суттєво більшою дисперсією, ніж

$$V^\pi(s_t) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t'=t}^{T} R(s_{t'}, a_{t'})$$

Але, так як другу оцінку на практиці отримати неможливо, бо для цього треба багато разів перегравати епізод з моменту часу t. Тобто, для цього потрібно мати можливість перемотки часу назад. Тому ми використовуємо першу більш шумну оцінку, що насправді показує досить хороші результати, незважаючи на дисперсію. В рамках цього методу, множина навчальних прикладів для параметричного апроксиматора V^{π} мають вигляд:

$$y_{i,t} = \sum_{t'=t}^{T} R(s_{i,t'}, a_{i,t'}),$$
$$\{(s_{i,t}, y_{i,t})\}$$

Тобто, мінімізується похибка:

$$L(\theta) = \frac{1}{2}\sum_{i} \left\| \hat{V}_{\theta}^{\pi}(s_i) - y_i \right\|^2$$

2. Бутстрап

$$V^{\pi}(s_t) \approx R(s_{i,t}, a_{i,t}) + \hat{V}_{\theta}^{\pi}(s_{i,t+1})$$

Бутстрап оцінка використовує оцінку на попередньому кроці оптимізації. Стабільність цієї оцінки не дуже очевидна, однак на практиці є дійсно ефективною. В рамках цього методу, множина навчальних прикладів для параметричного апроксиматора V^{π} мають вигляд:

$$y_{i,t} = R(s_{i,t}, a_{i,t}) + \hat{V}_{\theta}^{\pi}(s_{i,t+1}),$$
$$\{(s_{i,t}, y_{i,t})\}$$

Тобто, мінімізується похибка:

$$L(\theta) = \frac{1}{2}\sum_i \left\| \hat{V}_\theta^\pi(s_i) - y_i \right\|^2$$

2.3.2 Актор-Критик

Тепер, можна сформувати алгоритм навчання, що застосовує один із видів оцінювання, описаних в попередньому розділі. На наступній схемі (рисунок 2.12) описаний алгоритм, що використовує бутстрап оцінку:

Повторювати:

Згенерувати траєкторії $\{(s_i, a_i)\}$, використовуючи $\pi_\theta(a|s)$

Оновити ваги $\hat{V}_\theta^\pi(s)$ (1 методом)

$\hat{A}^\pi(s_i, a_i) = R(s_i, a_i) + \hat{V}^\pi(s_{i+1}) - \hat{V}^\pi(s_i)$

$\nabla_\theta J(\theta) \approx \sum_i \nabla_\theta \log \pi_\theta(a_i|s_i)\hat{A}^\pi(s_i, a_i)$

$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Рисунок 2.12 - Алгоритм Актор-Критик

У випадку дисконтування, алгоритм дещо зміниться. Його стабільність зростає, дисперсія оцінки градієнту зменшиться. На наступних схемах показані алгоритми із дисконтуванням в двох режимах: онлайн режим (рисунок 2.13) та батч режим (рисунок 2.14):

Повторювати:

Вибрати дію $a \sim \pi_\theta(a|s)$, отримати (s, a, s', R)

Оновити ваги $\hat{V}_\phi^\pi(s)$ (2 методом)

$\hat{A}^\pi(s_i, a_i) = R(s, a) + \gamma \hat{V}^\pi(s') - \hat{V}^\pi(s)$

$\nabla_\theta J(\theta) \approx \nabla_\theta \log \pi_\theta(a|s)\hat{A}^\pi(s, a)$

$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Рисунок 2.13 – Алгоритм онлайн Актор-Критик із дисконтуванням.

Повторювати:

Згенерувати траєкторії $\{(s_i, a_i)\}$, використовуючи $\pi_\theta(a|s)$

Оновити ваги $\hat{V}_\theta^\pi(s)$ (1 методом)

$\hat{A}^\pi(s_i, a_i) = R(s_i, a_i) + \gamma \hat{V}^\pi(s_{i+1}) - \hat{V}^\pi(s_i)$

$\nabla_\theta J(\theta) \approx \sum_i \nabla_\theta \log \pi_\theta(a_i|s_i) \hat{A}^\pi(s_i, a_i)$

$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Рисунок 2.14 - Алгоритм батч Актор-Критик із дисконтуванням

Актор-Критик загалом оцінює дві функції - функцію вартості та функцію стратегії. Апроксиматор функції стратегії називається Актор, а апроксиматор функції вартості - Критик.

2.7 Глибокий детермінований градієнт стратегій

Глибокий Детермінований Градієнт Стратегії (ГДГС) – це алгоритм, який одночасно вивчає Q-функцію та стратегію. Він використовує накопичені данні для навчання Q-функції за допомогою рівняння Белмана, і за допомогою Q-функції вивчає стратегію.

Для обмежених дискретних просторів дій, цей алгоритм не має великого сенсу, адже проблема максимізації Q-функції в такому випадку не викликає проблем. Проте, коли простір дій стає неперервним, проблема максимізація набуває дуже великої складності. ГДГС – це один із варіантів навчання Q-функцій для випадку неперервних дій.

Оскільки простір дій є неперервним, функція $Q^*(s, a)$ повинна бути диференційованою відносно дій. Це дозволяє вивести ефективне, градієнтне правило навчання для стратегії $\mu(s)$, яка використовує цей факт. Далі, замість

обчислювально складної оптимізаційної процедури для кожного обчислення $\max_a Q(s,a)$, ми можемо апроксимувати значення наступним чином:

$$\max_a Q(s,a) \approx Q\big(s,\mu(s)\big)$$

Як вже було описано в попередній параграфах, рівняння Белмана має наступний вигляд:

$$Q^*(s,a) = \mathbb{E}_{s'\sim P}[r(s,a) + \gamma\, max_{a'}\, Q^*(s',a')]$$

Це рівняння лежить в основі функціоналу середньої помилки Белмана, мінімізація якого дозволяє навчати нелінійні апроксиматори:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim Buffer}\left[\left(Q_\theta(s,a) - \left(r + \gamma \max_{a'} Q_\theta(s',a')\right)\right)^2\right]$$

В дискретному випадку, ми отримаємо алгоритм глибоких Q-мереж. Проте, важливо зазначити декілька важливих технік для стабілізації навчання.

По-перше, Буфер епізодів. Всі стандартні алгоритми глибокого навчання Q-функцій накопичують досвід в буфері, використовуючи його для подальших оновлень нейромереж. Це дозволяє апроксимувати динаміку середовища і стабілізує навчання.

По-друге, Цільові Мережі. Ця модифікація полягає у використанні окремої мережі для оцінювання $r + \gamma \max_{a'} Q_\theta(s',a')$. Це дозволяє поліпшити порушення стаціонарності даних навчання. Параметри цієї мережі періодично оновлюються параметрами основної мережі копіюванням або оновленням Поляка.

Таким чином, Глибокий Детермінований Градієнт Стратегії полягає у мінімізації наступного функціоналу:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim Buffer}\left[\left(Q_\theta(s,a) - \left(r + \gamma\, Q_{\theta_{target}}\left(s', \mu_{\theta^\mu}{}_{target}(s')\right)\right)\right)^2\right]$$

Крім того, для навчання функції детермінованої стратегії $\mu(s)$ в рамках ГДГС використовують градієнтний метод для рішення наступної задачі:

$$\max_{\theta^\mu} \mathbb{E}_{s\sim D}[Q_\theta(s, \mu_{\theta^\mu}(s))]$$

Тут параметри Q функції вважаються константними. Таким чином, навчання стратегії представляє собою досить просте градієнтне оновлення.

ГДГС вивчає детерміновану стратегію використовуючи попередній досвід. Через те, що стратегія є детермінованою, у агенту можуть виникнути проблем із дослідженням середовища. Щоби покращити процес дослідження в ГДГС, добавляють випадковий шум до дії під час навчання. Автори оригінальної статті по ГДГС рекомендують скорельований в часі випадковий процес Орнштейна-Уленбека. Цей процес визначається стохастичним диференціальним рівнянням:

$$dx_t = -\theta x_t dt + \sigma dW_t,$$

де $\theta > 0$ та $\sigma > 0$ – параметри, а W_t – Вінерівський процес.

2.8 Висновки

В даному розділі було проведено детальний огляд алгоритмів різних сімейств із глибокого навчання з підкріпленням. Спочатку надається класифікація та загальний огляд категорій та видів.

До описаних типів алгоритмів відносимо алгоритми, що явно використовують модель динаміки середовища, що загалом є більш ефективними в перерахунку на кількість прикладів для навчання. Серед цих алгоритмів виділено ті, що явним чином оцінюють функцію динаміки, та ті, які використовують вже існуючу апріорну інформацію про середовище та її динаміку.

Друга група – безмодельні алгоритми глибокого навчання з підкріпленням, що загалом є більш універсальними, та зачасту досягають кращих результатів, ніж алгоритми першої групи.

До другої групи відносять алгоритми типу Актор-Критик [41], які собою являють алгоритми градієнту стратегії [40-42] в яких оцінюється функція переваги. До її оцінки можна підійти з декількох сторін. В цій роботі описана оцінка на базі наближення функції вартості. Апросиматор цієї функції зветься Критиком. Процедура навчання може базуватись на двох типах оцінок: Монте-Карло та Бутстрап.

В кінці розділу надається опис алгоритму Глибокого Детермінованого Градієнту Стратегії [39]. Цей алгоритм розширює можливості глибокого Q-навчання, надаючи можливість застосування алгоритму для неперервних просторів дій. Він поєднує оновлення Q-функції та детермінованої функції стратегії. В ньому оцінювання апроксиматорів цих функцій відбувається по черзі, де функція стратегії навчається так, щоби максимізувати Q-функцію за діями, а Q-функція навчається, вважаючи, що використання отриманої поточної функції стратегії еквівалентно операції максимізації, наявної в функціоналі

похибки Белмана. Цей алгоритм надалі буде використовуватися в основі більш складної схеми навчання в запропонованому підході.

РОЗДІЛ 3 РОЗРОБКА СТРУКТУРИ ПРОГРАМНОГО КОМПЛЕКСУ

3.1 Обрані технології

Методи, які використовувалися для вирішення поставленої задачі використовують складений алгоритм, який попри все використовує глибокі нейронні мережі в якості нелінійних апроксиматорів. Тому, для реалізації потрібно було мати бібліотеку для глибокого навчання.

Через це, в якості мови програмування, був використаний Python та бібліотеки PyTorch та TensorFlow для глибокого навчання. Крім того, для обчислень використовувався сервер.

Крім того, для задач торгівлі цінними паперами використовувалася бібліотека TensorTrade, що описує основні примітиви для роботи із фінансовими рядами на основі даних про цінні папери.

3.2 Структура програмного продукту

Програмний продукт складається із трьох модулів.

Перший – модуль нейронних мереж, в якому були реалізовані нелінійні апроксиматори, що використовувалися в основному алгоритмі.

Другий – модуль алгоритму, який реалізовує алгоритм, що описаний нижче. Архітектурні рішення, що були прийняті під час розробки, спрямовані на підвищення універсальності модулю, та можливості інтеграції із TensorTrade.

Третій – модуль, що представляє собою симуляцію середовища та інструменти для інтеграції із TensorTrade.

3.3 Торгівля цінними паперами методами довгострокового планування

На сьогодні існує дуже велика кількість робіт, в яких задачу торгівлі цінними паперами вирішують за допомогою методів глибокого навчання з підкріпленням [16-18]. Проте, частіше за все, зусилля обмежуються використанням підходів, що не використовують модель. Наприклад, глибокі Q-мережі або наближена оптимізація стратегій. Попри те, що ці методи дають досить сильні результати, їх застосування в ситуаціях із великим горизонтом Марковського процесу прийняття рішень, вони частіше за все сходяться до локально оптимальних стратегій. Вони не мають змоги враховувати довгострокові закономірності ринку, а тому не мають великих переваг перед класичними методами, які потребують набагато менше обчислювальних ресурсів.

В подібних роботах задача торгівлі цінними паперами мала наступну форму:

$$A = [-1, 1]^N$$

$$S = \mathbb{R}^{6N+1}$$

$$r(s) = \text{приріст балансу}$$

На кожному кроці, агент приймає рішення щодо продажу або купівлі паперів кожної категорії. Це рішення кодується у вигляді числа, що означає частку паперів із максимального діапазону, що можливо купити або продати.

Простір станів описується вектором розмірності 6N+1. Для кожного виду паперів із N, надається 6 фінансових показників:

Price: поточна ціна закриття.

Shares: кількість акцій у портфелі.

MACD: Moving Average Convergence Divergence розраховується по ціні закриття.

RSI: Relative Strength Index (RSI) розраховується по ціні закриття.

CCI: Commodity Channel Index (CCI) розраховується по найвищій, найнижчій ціні та ціні закриття.

ADX: Average Directional Index (ADX) розраховується по найвищій, найнижчій ціні та ціні закриття.

Крім того, до вектору додається показник поточного балансу агенту.

Метрикою якості методів в даному випадку, будемо вважати баланс агенту. Причиною тому є бажання порівняти існуючі підходи із представленим в роботі, в той час як велика кількість із існуючих представлених результатів використовують саме баланс.

3.3.1 Моделі на основі темпоральних різниць

Основним нововведенням в даній роботі є використання прийомів та методів, що спроектовані для вирішення проблем з довгими строками. Ми розглядаємо ті з них, що діють в Марковських процесах прийняття рішень із проміжними цілями. В рамках цієї моделі, стратегії залежать від кінцевого стану, до якого треба прийти.

Розглянемо один з таких методів – моделі на основі темпоральних різниць. Основною ідеєю є об’єднання переваг підходів навчання з підкріпленням з та без моделей. В той час як перші надають можливість більш швидкого навчання за допомогою оцінювання моделі динаміки, другі - показують кращі результати в сенсі кінцевої метрики, проте потребують набагато більше часу для збіжності.

Ми вводимо тип цільових функцій цінності, які називаються тимчасовою різницею моделі, які забезпечують пряме з'єднання з RL на основі моделі. Спочатку ми мотивуємо цей зв’язок, пов’язуючи оптимізацію з цільовими умовами функції значення.

Розглянемо вибір функції винагороди для умовної за ціллю функції вартості. Хоча у літературі досліджено різноманітні варіанти, наприклад, зв’язок із навчанням з підкріпленням із моделями виникає, якщо ми множина цілей G = S, така що g ∈ G відповідає цільовому стану s_g ∈ S, і ми вважаємо, що функція винагороди в МППР базується на відстані до цільового стану наступним чином:

$$r_d(s_t, a_t, s_{t+1}, s_g) = -D(s_{t+1}, s_g),$$

де $D(s_{t+1}, s_g)$ - відстань, наприклад, евклідова: $D(s_{t+1}, s_g) = \|s_{t+1} - s_g\|_2$.

Якщо γ = 0, маємо $Q(s_t, a_t, s_g) = -D(s_{t+1}, s_g)$ при збіжності Q-функції, що означає, що із $Q(s_t, a_t, s_g) = 0$ слідує, що $s_{t+1} = s_g$. Підставимо цю Q функцію до рівняння оптимізації з моделлю, позначаючи оригінальну функцію винагороди через r_c.

$$a_t = argmax_{a_{t:t+T}, s_{t+1:t+T}} \sum_{i=t:t+T} r_c(s_i, a_i). \quad \text{За умови } Q(s_t, a_t, s_{i+1}) = 0$$
$$\forall i \in \{t, .., t+T-1\}$$

Рішення цієї задачі і є планом заснованим на моделі. Таким чином ми вивели точний зв'язок між безмодельним навчанням з підкріпленням та заснованим на моделі. Тут можна використовувати безмодельне навчання цільових обумовлених функцій вартості, щоби безпосередньо описати неявну модель, яка може бути використана для планування. Однак цей зв’язок сам по собі не дуже корисний: отримана неявна модель повністю базується на моделі динаміки і не надає ніяких можливостей для вирішення проблем довгострокового планування. Далі ми покажемо, як розширити модель, вводячи поняття темпоральних різниць.

Якщо розглянути випадок, коли γ > 0, задача оптимізації, описана вище, більше не відповідає жодному методу оптимального керування. Насправді, коли

$\gamma = 0$, значення Q функції мають чітко визначені одиниці виміру: одиниці відстані між станами. При $\gamma > 0$ таке тлумачення неможливе. Ключовим моментом у моделях темпоральних різниць є запровадження іншого механізму агрегування довгого горизонту винагород. Замість оцінки значень Q як дисконтованих сум винагород введемо додатковий параметр τ, який представляє горизонт планування. Тепер рекурсивне рівняння Белмана для Q-функції буде виглядати як:

$$Q(s_t, a_t, s_g, \tau) = \mathbb{E}_{p(s_{t+1}|s_t,)}[-D(s_{t+1}, s_g)\mathbb{I}[\tau = 0] + \\ + \max_a Q(s_{t+1}, a, \tau - 1)\mathbb{I}[\tau \neq 0]].$$

Q-функція використовує винагороду $-D(s_{t+1}, s_g)$, коли $\tau = 0$ (на цьому епізод закінчується), і зменшує τ на одиницю на кожному наступному кроці. Оскільки це все ще чітко визначена рекурсія Q-навчання, її можна оптимізувати за допомогою даних, і, як і для умовних цільових функцій вартості, ми можемо вибрати нові цілі s_g та нові горизонти τ для кожного кортежу (s_t, a_t, s_{t+1}), навіть тих, які не були фактично наявними під час збору даних. Таким чином, моделі з темпоральними різницями можна навчити дуже ефективно, оскільки кожен кортеж забезпечує сигнал навчання за усіма можливими цілями та кожним можливим горизонтом.

Інтуїтивна інтерпретація моделей за темпоральними різницями полягає в тому, що метод повідомляє нам, наскільки близько агент наблизиться до даного цільового стан s_g після τ кроків часу, коли він намагається досягти цього стану за τ кроків. Як варіант, результат можна інтерпретувати як значення Q функції у кінцевому горизонті МППР, де горизонт визначається параметром τ. У випадку, коли $\tau = 0$, метод ефективно вивчає модель, що дозволяє інтегрувати його у різноманітні схеми планування та оптимального керування. Таким чином, ми можемо розглядати метод моделей за темпоральними різницями як інтерполяцію між модельним та безмодельним навчанням з підкріпленням, де значення

параметру $\tau = 0$ відповідає однокроковому прогнозу, зробленому в рамках модельного навчання, а $\tau > 0$ відповідає довгостроковому прогнозуванню, зробленому звичайними Q-функціями. Попри те що ця відповідність зміниться, для $\tau > 0$, якщо користувача цікавить лише винагороду на кожному K-тому кроці, тоді ми можемо переписати відповідність наступним чином:

$$a_t = argmax_{a_{t:K:t+T}, s_{t+K:K:t+T}} \sum_{i=t,t+K,\ldots,t+T} r_c(s_i, a_i).$$

$$\text{За умови } Q(s_t, a_t, s_{i+K}, K-1) = 0$$

$$\forall i \in \{t, t+K, .., t+T-K\}$$

де ми оптимізуємо лише для кожного K-того стану і дії. Оскільки метод набуває ефективності при довших горизонтах, ми можемо збільшувати K до K = T і планувати лише на один ефективний часовий крок:

$$a_t = argmax_{a_t, a_{t+T}, s_{t+T}} r_c(s_{t+T}, a_{t+T}).$$

$$\text{За умови } Q(s_t, a_t, s_{i+T}, K-1) = 0$$

Це формулювання призводить до певної втрати загальності, оскільки ми більше не оптимізуємо винагороду за проміжні кроки. Це обмежує багатоступеневе формулювання проблемами кінцевих (розріджених) винагород, але це дозволяє нам обробляти довільні функції винагороди на кінцевому стані s_{t+T}, що все ще описує широкий спектр практично актуальних задач.

В якості алгоритму навчання з підкріпленням в основі методу, ми використали метод глибоких детермінованих градієнтів стратегії.

3.4 Висновки

В цьому розділі було описано процес і деталі проектування програмного продукту. Проектування для даної роботи включало в себе багато різноманітних під-задач. Основними з них були забезпечення інтеграції із симулятором торгівлі на ринку цінних паперів, а також реалізація описаного в даному розділі підходу та алгоритмів. Основними труднощами були ті, що пов’язані із реалізацією досить складного алгоритму, що включає в себе навчання із підкріпленням та роботу із буфером епізодів, які агент зберігає для подальшого використання для покращення своєї роботи.

РОЗДІЛ 4 АНАЛІЗ ПРАКТИЧНОГО ДОСЛІДЖЕННЯ

4.1 Дані

В даній роботі для експериментів та порівняння був використаний набір даних по цінних паперах із Wharton Research Data Services (WRDS). Там було використано 30 паперів із Dow Jones, та використано історичні дані із 01/08/2015 по 09/08/2020 для навчання агенту та визначення якості його роботи. Рішення приймається один раз на день по кожній позиції портфеля.

На цьому наборі даних ми порівнюємо декілька існуючих підходів до автоматичної торгівлі цінними паперами із методом, запропонованим в даній роботі.

4.2 Результати та порівняння

Спершу, розглянемо процесс навчання та підбір параметрів алгоритму моделей із темпоральними різницями. На рисунку 4.1 можна побачити графіки функції похибки для різних значень максимального горизонту Марковського процесу прийняття рішень із проміжними цілями.

Як видно, найкращим значенням виявилося $\tau = 15$. Коли значення цього параметру дуже мале, модель втрачає можливість відображати довгі залежності в даних. В той час, як більші значення можуть привести до підвищення дисперсії оцінок градієнтів в алгоритмі Глибокого Детерміновного Градієнту Стратегії, через що процес навчання втрачає стабільність.

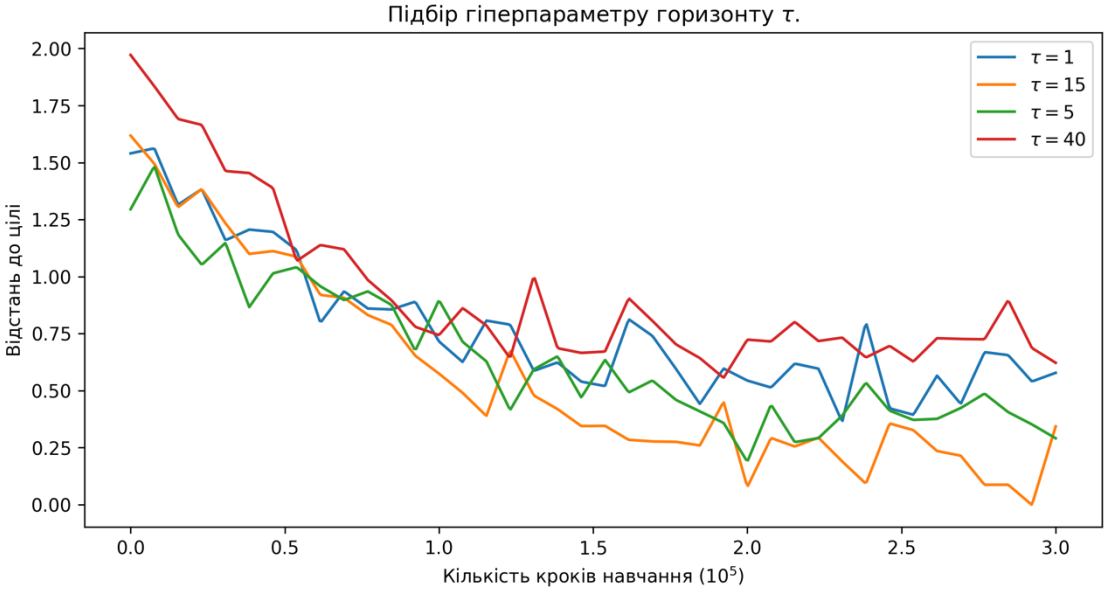

Рисунок 4.1 – Залежність похибки алгоритму моделей із темпоральними різницями від довжини горизонту.

Вибравши найкраще значення, вибірку було розділено на дві підвибірки (рис 4.2), що не перетинаються. На рисунку 4.3 зображена еволюція похибки на навчальній та тестувальній підвибірках.

Рисунок 4.2 – розбиття часових рядів на підвибірки. Тренувальна вибірка – 2015/01/08 – 2017/01/09, валідаційна підвибірка – 2017/02/09 – 2018/01/08 та

відкладена вибірка, що використовується при порівнянні – 2018/01/08 – 2020/09/08.

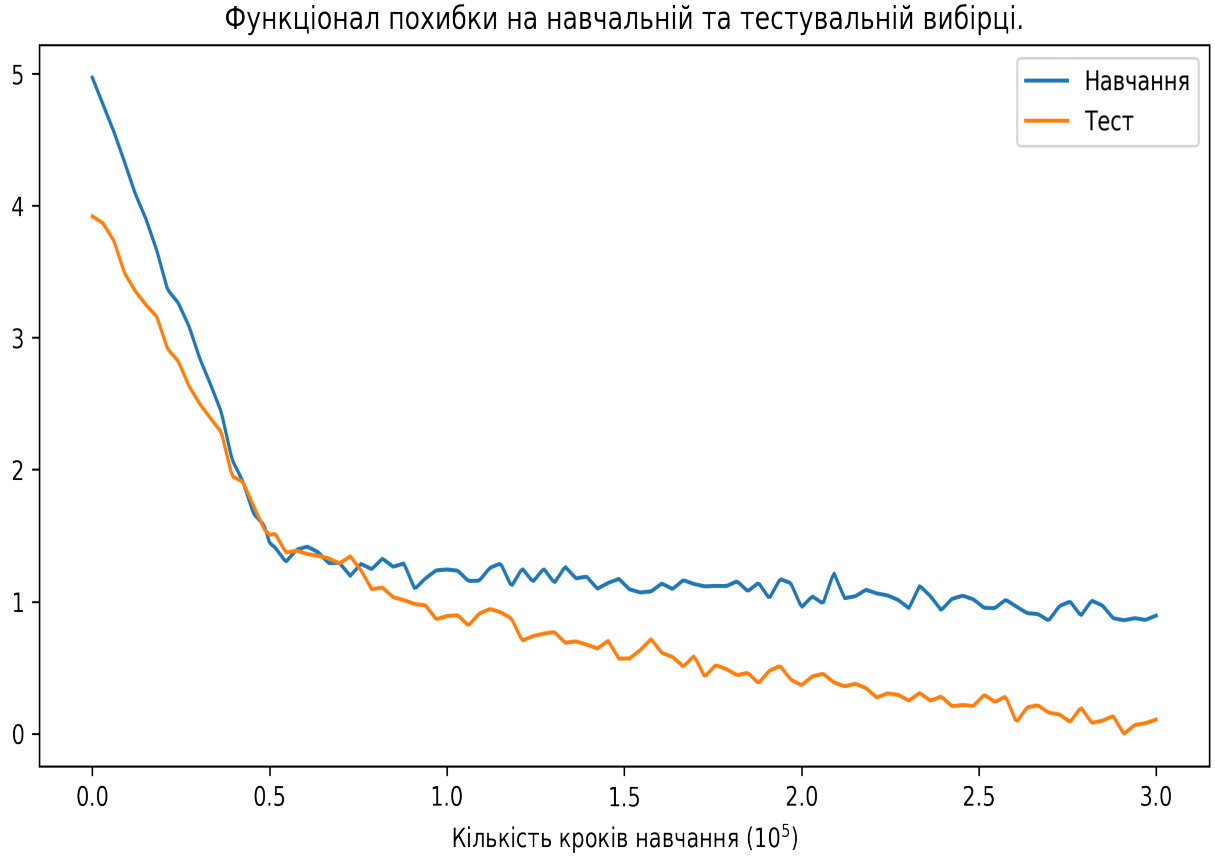

Рисунок 4.3 – графік функції похибки під час процесу навчання на тренувальній та валідаційній вибірках. Зауважимо, що похибка на валідаційній вибірці є меншою через меншу варіативність розподілу данних в ній.

Тепер розглянемо результати порівняння методів. Крім основного методу, в перелік алгоритмів були включені Наближена Оптимізація Стратегії (PPO), Глибокий Детермінований Градієнт Стратегії (DDPG) та Ассинхронний Актор-Критик (A2C). Крім того, запропонований метод поданий у двух варіантах: той, що базується на алгоритмі Глибокого Детермінованого Градієнту Стратегії та той, що базується на його більш стабільній модифікації - алгоритмі Глибокого Детермінованого Градієнту Стратегіх із Подвійною затримкою. В останньому використовуються методи подвійного Q-навчання, сглажування цільової стратегії та затриманого оновлення стратегії.

На наступному графіку (рис 4.4) зображені результати порівняння інсуючих автоматичних методів торгівлі цінними паперами. Серед них класичні методи торгівлі такі, як Min-Variance та DJIA, а також методи, засновані на методах глибокого навчання із підкріпленням без моделей. Крім того, для порівняння представлені результати, отримані описаним в даній роботі методом.

Із нього видно, що запропонований метод співпадає із існуючими на коротких епізодах торгівлі, проте показує значно кращі результати на епізодах великої довжини за рахунок виявлення та урахування довгострокових залежностей у часових рядах. Як видно, за деякий проміжок часу, отримана стратегія реєструє важливі довгострокові залежності, що додають апріорних знань при подальших прогнозуваннях.

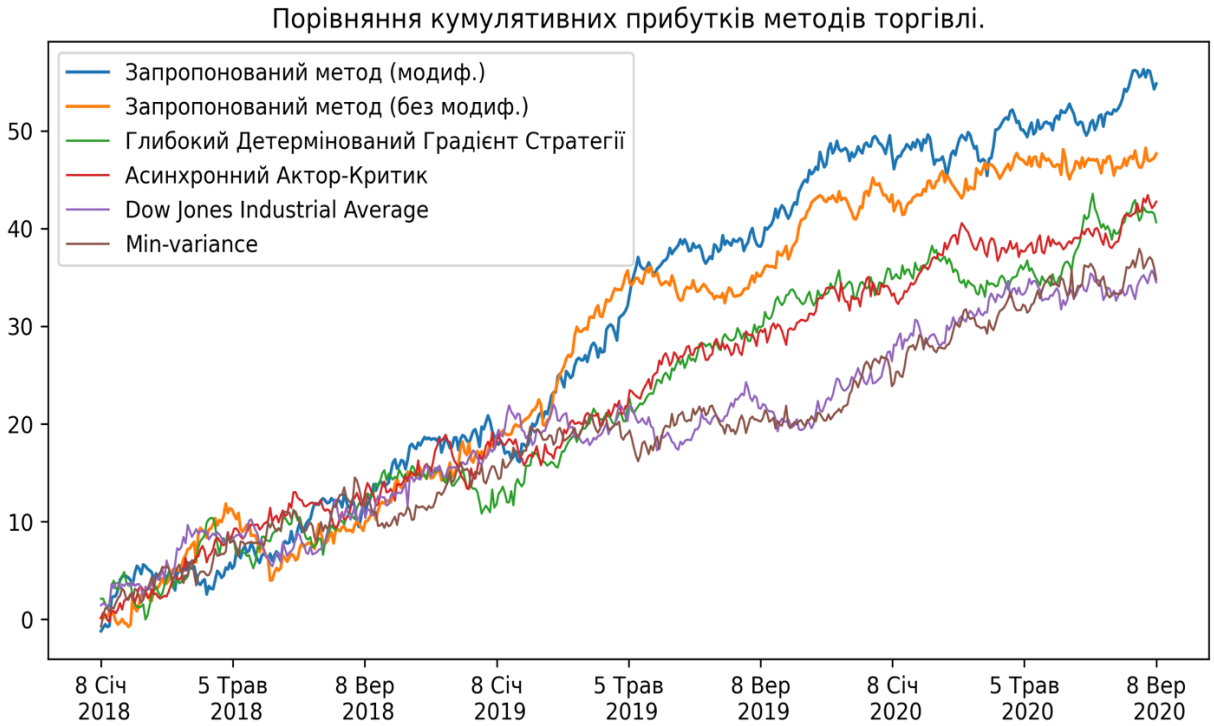

Рисунок 4.4. Порівняння поточних балансів серед методів автоматичної торгівлі цінними паперами. Синім позначений запропонований алгоритм із використанням модифікації ГДГС – TD3.

На наступній таблиці (табл. 4.1) зображені кінцеві показники результатів роботи стратегій, навчених різними алгоритмами глибокого навчання з підкріпленням та класичних методів автоматичної торгівлі цінними паперами. Зауважимо, що на даному наборі даних

Таблиця 4.1. Порівняння результатів роботи алгоритму. Порівняння відбувається за 5 показниками: кінцевий баланс (кумулятивна винагорода), річний прибуток та волатильність, відношення Шарпе та розмір максимального провалу.

2018/01/08-2020/09/08	**Запроп. метод (модиф.)**	**Запроп. метод**	**Асинхронний Актор-Критик**	**Глибокий Детермінований Градієнт Стратегії**	**DIJA**	**Min-Variance**
Кумулятивний прибуток	54%	49%	41%	41%	34%	31%

Продовження таблиці 4.1

Щорічний прибуток	11.4%	10.1%	9.1%	9.5%	7.8%	7.2%
Щорічна волатильність	9.6%	9.3%	8.4%	11.3%	17.8%	20.1%
К-т. Шарпе	1.012	0.959	0.91	0.83	0.46	0.49
Максимальне падіння	-8.1%	-6.1%	-4.3%	-5.5%	-6.3%	-15%

4.2.1 Аналіз отриманих стратегій

В попередньому розділі було наведено порівняння підходів глибокого навчання з підкріпленням та класичних методів. Аналогічний аналіз був

проведений в [43-44]. В нашому випадку, на модельних даних ми порівняли запропонований метод із та без модифікації з іншими методами. Стратегії, отримані іншими методами також були проаналізовані в роботах [43-44].

У випадку запропонованого методу, стратегії статистично (середній час транзакції, максимальний розмір транзакції та максимальний поточний розмір портфелю) майже не відрізняються. Перевагою нашого підходу є довший період прибутковості. В той час як, у [44] перетренування моделі відбувається кожні 5 років при розмірі навчальної вибірки у 10 років, та 1 рік при розмірі навчальної вибірки у 4 роки, запропонований метод виходить на плато за 2-3 роки при розмірі вибірки у 2 роки. Такий ефект досягається через використання Моделей із Темпоральними Різницями [19] в основі методу.

Крім того, на модельних даних, запропонований метод досягає більшої прибутковості при фіксованій ціні транзакції (короткі та довгі).

4.3 Висновки

В цьому розділі було представлено і проаналізовано дані та результати роботи запропонованого підходу до торгівлі цінними паперами за допомогою навчання із підкріпленням.

Був представлений опис експерименту, принцип поділу даних на підвибірки. Було продемонстровано процес підбору гіперпараметру, а також графіки, що описують процес навчання.

Із результатів випливає, що запропонований метод на базі Моделей із Темпоральними Різницями [19] і модифікації ГДГС [42] співпадає із існуючими на коротких епізодах торгівлі, проте показує значно кращі результати на епізодах великої довжини за рахунок виявлення та урахування довгострокових залежностей у часових рядах. Серед методів для порівняння були взяті Асинхронний Актор-Критик [16], ГДГС [39] та класичні методи.

Таким чином, даний алгоритм можна використовувати як автоматичний торгівельний робот, а також модель ринку, що досить локально, проте точно відображає динаміку в часових рядах. Серед недоліків методу, можна виділити складність реалізації та вихід на плато по балансу, коли стратегія перестає приносити суттєвий прибуток.

Крім того, у порівнянні з [44] ефективність методу виявляється більшою, адже період перетренування, необхідного для актуалізації моделі, виявляється довшим при фіксованому розмірі навчальної вибірки.

На практиці, системи автоматичної торгівлі цінними паперами мають набагато складнішу структуру. Наприклад, метод періодично може перетреновуватися на нових даних або система може вибирати метод в залежності від оцінки ситуації, або складності обчислень.

Подолання цього недоліку не розглядається в рамках даної роботи. Це питання представляє можливість для більш глибокого дослідження. Тому, залишаємо цей напрям для подальшої роботи.

РОЗДІЛ 5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

В останні роки набув великої популярності такий вид малого підприємництва як стартап. Стартап проект – є комерційним проектом, який знаходиться в стані розробки, або нещодавно вийшов на ринок. Характерною особливістю стартапу, що відрізняє його від малого бізнесу, є оригінальність та інновації, він не може бути копією вже реалізованих ідей. При цьому проект не обов'язково повинен бути масштабного характеру, головне, щоб він був креативним, а його завдання – спрощувати людям будь-які дії в їх повсякденному житті. Наразі, з появою Інтернету та сучасних технологій, стало простіше заходити на ринок, знаходити інвесторів та споживачів. З'явилося набагато більше можливостей для розвитку свого проекту за кордоном, ніж раніше. Проте розробка стартапу є досить ризикованим завданням. Не всім вдається довести свій стартап проект до ринкового впровадження. За статистикою успіху досягає лише $10 - 20\%$ від усіх стартап проектів. Запуск стартапу передбачає цілий ряд обов'язкових дій, в межах яких визначають ринкові перспективи стартапу, графік розробки, принципи організації виробництва, заходи з залучення інвесторів та аналіз ризиків.

5.1 Опис ідеї стратап проекту

Назва стартап проекту: "Система для трейдингу з довгостроковим плануванням".

У таблиці 5.1 подано зміст ідеї стартап проекту, можливі напрямки застосування та основні вигоди, що може отримати користувач товару. У таблиці 5.2 визначені сильні, слабкі та нейтральні сторони проекту.

Таблиця 5.1 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система для трейдингу, що враховує далеке минуле, а також має великий горизонт планування. Ця система допомагає трейдеру приймати рішення щодо купівлі/продажу предмету тогргівлі	Використання системи для трейдингу, що враховує далеке минуле, а також має великий горизонт планування трейдерами.	Система для трейдингу, що враховує далеке минуле, а також має великий горизонт планування, надаючи більш розумні рішення щодо купівлі/продажу.

Таблиця 5.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Технічно-економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів		
		Мій проект	TensorTrade	Open-Trader
1	Ціна	Низька	Висока	Низька
2	Функціонал	Широкий	Широкий	Вузький

Отже, з табл. 5.2 можна визначити, що ціна та функціонал є сильними характеристиками для потенційного товару.

5.2 Технологічний аудит ідеї проекту

За результатами аналізу таблиці 5.3 можна зробити висновок про можливість технологічної реалізації проекту.

Таблиця 5.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Система для трейдингу, що враховує далеке минуле, а також має великий горизонт планування.	Навчання з підкріпленням на основі нейронних мереж (Python, TensorFlow, MongoDB, C++, AngularJS).	Наявна	Доступна
2.	Система для високочастотного трейдингу, що враховує далеке минуле, а також має великий горизонт планування	Навчання з підкріпленням з наявними прискорювачами (Python, C++, XGBoost, SQLServer).	Наявна	Недоступна
Обрана технологія реалізації ідеї проекту: Навчання з підкріпленням на основі нейронних мереж (Python, TensorFlow, MongoDB, C++, AngularJS).				

5.3 Аналіз ринкових можливостей запуску стартап проекту

5.3.1 Аналіз попиту та потенційних груп клієнтів

Визначення ринкових можливостей (табл. 5.4), які можна використати під час ринкового впровадження проекту та ринкових загроз, які можуть перешкодити реалізації проекту дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів конкурентів.

Таблиця 5.4 - Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	15
2	Загальний обсяг продаж, грн/ум.од	100 000 ум.од.
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає обмежень для входу
5	Специфічні вимоги до стандартизації та сертифікації	Немає специфічних вимог до стандартизації та сертифікації
6	Середня норма рентабельності в галузі (або по ринку), %	75 %

Середня норма рентабельності галузі вища за банківський відсоток на вкладення, що означає привабливість ринку для входження за попереднім оцінюванням.

Визначимо потенційні групи клієнтів, їх характеристики, та сформуємо орієнтовний перелік вимог до товару для кожної групи (табл. 5.5).

Таблиця 5.5 - Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Потреба підтримки прийняття рішень для трейдерів	Нові компанії, котрі не мають своєї системи підтримки прийняття рішень, або ж компанії чиї системи є неефективними.	Клієнт використовує усі можливості системи, як результат отримує кінцевий оптимальне рішення.	- Простота у використанні. - Висока точність прогозування.
2.	Потреба в урахуванні складних залежностей для прийняття рішень для трейдерів	Компанії, котрі уже мають певний підхід до автоматичного трейдингу, однак потребують більшої точності прогнозу, так як попереднє прогнозування не існує або не є ефективним.	Клієнту надається лише частина продукту, а саме ПЗ, що відпоідає за прогнозування.	- Зручний інтерфейс. - Ефективний підхід до прогнозування. - Швидкодія (великі об’єми даних)

5.3.2 Аналіз ринкового середовища

Проведемо аналіз ринкового середовища: таблиці факторів, що сприяють ринковому впровадженню проекту та факторів, що йому перешкоджають (табл. 5.6 - 5.7).

Таблиця 5.6 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Конкуренція	Реліз нової версії системи TensorTrade з вдосконаленою аналітикою для користувачів.	Вдосконалити або модифікувати алгоритм та аналітику платформи.
2.	Постачальники	Постачальники планують підвищити ціну на ліцензійне програмне забезпечення, необхідне для реалізації та підтримки проекту.	Залучити інвестиції з урахуванням змін, вести перемовини з постачальниками з приводу фіксування ціни за умов довготривалої співпраці
3.	Збут	Лояльність споживачів до існуючих конкурентів може призвести до низької зацікавленості у власному продукті.	Потрібно розробити вдалі маркетингову та рекламну кампанії.

Продовження таблиці 5.6

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
4.	Зміна потреб користувача	Користувачам необхідні рішення з іншим функціоналом.	Передбачення можливості додавання нового.

Таблиця 5.7 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Конкуренція	Збільшення цін на тарифи користування системою TensorTrade.	Демпінгувати ціни на послуги побудови рекламних кампаній.
2.	Постачальники	Поява нових постачальників, з більш вигідними пропозиціями ресурсів.	Налагодити комунікацію з потенційними партнерами.
3.	Збут	Великі компанії конкуренти планують підвищити ціну за свої послуги.	Розглядати стратегію демпінгування ціни.
4.	Поява нових цільових груп клієнтів	Потреба в аналогічному продукті в інших сферах діяльності.	Адаптація продукту під нові сфери використання.

5.3.3 Аналіз пропозиції

Проведемо аналіз пропозиції: визначимо загальні риси конкуренції на ринку (табл. 5.8).

Таблиця 5.8 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополістична конкуренція	Ринок складається з відносно великої кількості продавців, кожен з яких володіє невеликою (але не нескінченно малою) часткою ринку; встановлюючи ціни, продавці намагаються виділитися за неціновими ознаками; товар кожного продавця є недосконалим замінником товарів інших фірм; ринок не має бар'єрів для входу та виходу	Для того, щоб відрізнятися від конкурентів чимось, окрім ціни, продавці розробляють різні пропозиції для різних сегментів споживачів і широко використовують практику використання марок, рекламу та методи особистих продажів. Також необхідною є підтримка якості продукту та постійні вдосконалення.
2. За рівнем конкурентної боротьби - інтернаціональний рівень	Конкуренти діють на інтернаціональному просторі - в мережі Інтернет	Забезпечити можливість користування послугами (продуктом) незалежно від місцезнаходження. Підтримувати продукт на національному ринку
3. За галузевою ознакою - внутрішньогалузева	Має місце суперництво між окремими підприємцями і фірмами однієї галузі щодо одержання прибутку.	Підвищення продуктивності праці, зменшення витрат виробництва, зниження індивідуальної цінності товару, розширення функціоналу продукту, вдосконалення продукту для застосування у нових галузях (для нових видів транспортів чи перевзок)
4. Конкуренція за видами товарів: - товарно-видова	Це конкуренція між товарами одного виду - рідні рекомендаційні системи виконують один і тож же набір функцій	Використовувати цінові та неціновіметоди конкурентної боротьби на ринку

Продовження таблиці 5.8

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
5. За характером конкурентних переваг - нецінова	Проводиться головним чином за допомогою вдосконалення якості продукції, технології виробництва, інновацій та нанотехнологій, патентування і брендування і умов її продажу	Випускати новий товар (послугу), які або принципово відрізняються від своїх попередників або які представляють модернізований варіант старої моделі
6. За інтенсивністю - не марочна	роль торгової марки незначна, хоча самі марки можуть бути присутніми на ринку	Немає необхідності вкладати кошти у створення та розкрутку бренду, необхідно приділяти увагу якості продукту а не бренду компанії

Після аналізу конкуренції проведемо більш детальний аналіз умов конкуренції в галузі (за моделлю 5 сил М. Портера) (табл. 5.9).

Таблиця 5.9 - Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	TensorTrade, Open-trade	AlgoTrade	-Ціна на ресурси - Готовність до співпраці -Якість ресурсів	- Вплив покупців на ринкову ціну - Очікування щодо якості обслуговування	Поява відносно дешевих або безкоштов-них замінників
Висновки	З боку прямих конкурентів очікується інтенсивна конкурентна боротьба.	- є можливість виходу на ринок, бар'єри відсутні - загалом названі вище компанії мають декілька вже реалізованих	Постачальники контролюють ціни на ресурси та їхню якість	Клієнти диктують умови роботи на ринку	Є схожі товари, однак дана версія є унікальною в певному році.

Продовження таблиці 5.9

Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Висновки	варантів схожої продукції та готують інновації, тому потенційний ріст популярності продукції компаній невідомий (може бути як і швидкий так і не дуже)			зараз не існує.

Отже, з огляду на конкурентну ситуацію у галузі, бачимо, що є можливість входження на ринок. Щоб бути конкурентноспроможним на ринку, проект має бути привабливим за ціною, якістю роботи та мати зручний інтерфейс. Зокрема проект дещо відрізняється від вже існуючих систем, а саме враховує дуже великий горизонт залежностей.

5.3.4 Фактори конкурентоспроможності

У таблиці 5.10 наведено фактори конкурентноспроможності, сформовані на основі аналізу конкуренції та з урахуванням характеристик ідеї проекту (табл. 5.9), вимог споживачів до товару (табл. 5.5) та факторів маркетингового середовища (табл. 5.6-5.7)

Таблиця 5.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Технологія	Рівень застосовуваних технологій відіграє значну роль в побудові ефективної системи трейдингу - а отже надання якісного сервісу споживачу.
2	Ціна	Ціна має значення як для приватнох особи - користувача онлайн-системи, так і для великих компаній, які зацікавлені у придбанні абонементу на користування ефективною системою трейдингу.
3	Маркетинг	• Частка ринку, займаного підприємством • Престиж торгової марки (бренд) • Витрати на стимулювання збуту і їх ефективність
4	Репутація	• Репутація (імідж) підприємства • Якість обслуговування

Продовження таблиці 5.10

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
5	Унікальність позиціонування	В умовах монополістичної конкуренції, коли фактор диференціації торгової марки є ключовим засобом ведення конкурентної боротьби, важливим є створення та підтримання унікального позиціонування, що створює певний захист від конкурентних зіткнень. Для поточного продукту наявна інновація, котра відрізняє його від вже існуючих продуктів на ринку.

5.3.5 Аналіз сильних та слабких сторін стартап-проекту

У таблиці 5.11 наведено аналіз сильних та слабких сторін проекту, сформульованих за визначеними факторами конкурентоспроможності (табл. 5.10).

Компанії-конкуренти:

1. TensorTrade
2. Open-trade
3. AlgoTrade

Таблиця 5.11 - Порівняльний аналіз сильних та слабких сторін «Система побудови рекламних кампаній»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні із товаром стартапу						
			–-3	*–-2*	*–-1*	*00*	*+1*	*+2*	*+3*
1	Технологія	20				1	1	2	3
2	Ціна	10				3		1 2	3
3	Маркетинг	9	1 2	3					
4	Репутація	8		1		2 3			
5	Унікальність	19					1	2 3	

5.3.6 SWOT-аналіз

У таблиці 5.12 наведено SWOT-аналіз (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 5.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 5.11).

Таблиця 5.12 - SWOT- аналіз стартап-проекту

Сильні сторони:	**Слабкі сторони:**
1. Ціна нижча за ціну конкурентів, 2. Унікальне позіціонування - як в якості відокремленого ПЗ для прогнозування так і в якості системи для трейдингу. 3. Наявна інновація, що впливає на ефекивність системи	1. Технологія відстає від ринкових лідерів, 2. Відсутність чітко вираженої маркетингової стратегії, непослідовність в її реалізації, 3. нейтральна репутація виробника.
Можливості:	**Загрози:**
1. Можливість зміцнення іміджу підприємства за допомогою реклами та маркетингу 2. Можливість перетворення головних конкурентів на партнерів (споживачів) - шляхом постачання абонементу на користування системою та надання ефективних рекомендацій.	1. Загроза працювати без прибутку внаслідок демпінгування цін 2. Загроза втрати споживачів внаслідок підвищення конкуренції з боку великих компаній-конкурентів в галузі 3. Загроза підвищення цін на користування послугами внаслідок підвищення цін на ресурси - як результат втрата покупців залучених низькими цінами.

На основі SWOTаналізу визначимо альтернативи ринкової поведінки (перелік заходів) для виведення стартаппроекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (табл. 5.13).

Таблиця 5.13 - Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Впровадження існуючого товару на існуючому ринку	Середня	4 міс
2.	**Розвиток товару на існуючому ринку**	**Висока**	**6 міс**
3.	Розвиток нового ринку з існуючим товаром	Низька	5 міс
4.	Впровадження нового товару і створення нового ринку	Дуже низька	8 міс

Обрано альтернативу 2, для якої ймовірність отримання ресурсів є високою і строк реалізації відносно невеликий.

5.4 Розроблення ринкової стратегії проекту

5.4.1 Вибір цільових груп потенційних споживачів

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 5.14).

Таблиця 5.14 - Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Компанії, котрі уже мають певний підхід до трейдингу, однак потребують більшої точності прогнозу.	Низька	Низький	Висока	Середня

Продовження таблиці 5.14

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
2.	Нові компанії, котрі не мають своєї системи трейдингу або ж компанії чиї системи трейдингу є неефективними.	Висока	Високий	Середня	Висока
Які цільові групи обрано: Нові компанії, котрі не мають своєї системи трейдингу.					

5.4.2 Базова стратегія розвитку

У таблиці 5.15 наведено базову стратегію розвитку.

Таблиця 5.15 - Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
	Розвиток товару на існуючому ринку	Стратегія концентрованого маркетингу.	Позиціонування за співвідношенням "ціна – якість", позиціонування на основі порівняння товару фірми з товарами конкурентів	Стратегія спеціалізації

5.4.3 Стратегія конкуренної поведінки

У таблиці 5.16 наведено стратегію конкурентної поведінки.

Таблиця 5.16 - Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки*
	Ні	Забирати існуючих у конкурентів	Так, з додаванням унікальних властивостей, суттєвим покращеням ефективності, характеристик продукту.	Стратегія наслідування лідера.

5.4.4. Стратегія позиціонування

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. табл. 5.5), а також в залежності від обраної базової стратегії розвитку (табл. 5.15) та стратегії конкурентної поведінки (табл. 5.16) розробляється стратегія позиціонування (табл. 5.17). що

полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 5.17 - Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
	Зручність у використанні (наявність простого та зручного користувацького інтерфейсу), ефективність, швидкість роботи, ціна .	Стратегія спеціалізації.	Ефективний трейдинг заснований на довгостроковому урахуванню залежностей.	Позиціювання на основі порівняння товару фірми з товарами конкурентів, Позиціювання на позитивних особливостях технології, Позиціювання за показниками якості.

5.5 Розроблення маркетингової програми стартап-проекту

5.5.1 Ключові переваги концепції потенціного товару

У таблиці 5.18 наведено результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.18 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Потреба в системі трейдингу.	Система планує на великий горизонт.	Більш точні прогнози, оптимальніші дії, довші та складніші стратегії.

5.5.2 Трирівнева маркетингова модель товару

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 5.19).

Таблиця 5.19 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
І. Товар за задумом	Система для трейдингу, що враховує далеке минуле, а також має великий горизонт планування. Ця система допомагає трейдеру приймати рішення щодо купівлі/продажу предмету тогргівлі.

Продовження таблиці 5.19

II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Система для трейдингу, що враховує далеке минуле, а також має великий горизонт планування.	N/A	N/A
	Якість: тестування, дослідження ефективності прогнозування з використанням декількох метрик (показників) якості, дослідження ефективності трейдингу.		
	Пакування: N/A		
	Марка: “SmartTrade”		
III. Товар із підкріпленням	До продажу: дослідження ефективності, тестування		
	Після продажу: онлайн-підтримка, гарантійне обслуговування, навчання персоналу		
За рахунок чого потенційний товар буде захищено від копіювання: Ліцензія. Вихідний код програмного продукту є закритим, та не передається клієнтам і третім особам. На програмний продукт оформлено авторське право			

5.5.3 Визначення цінових меж

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (табл. 5.20). Аналіз проводиться експертним методом.

Таблиця 5.20 - Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
	2000$ / рік	1000-1500$ / рік	90000$ / рік	Базова покупка та впровадження: нижня межа 1000$, верхня межа 2000$.

5.5.4 Формування системи збуту

У таблиці 5.21 наведено формування системи збуту.

Таблиця 5.21 - Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	Цільові клієнти – трейдери, трейдингові компанії, системи підтримки рішень для торгівлі	Формування попиту і стимулювання збуту. Встановлення контактів із споживачами. Просування маркетингової інформації. Забезпечення безпечної грошової транзакції.	Нульова або однорівнева (сервіс безпосередньо продається споживачам та через посередників)	Власні засоби збуту

5.5.5 Концепція маркетингових коммунікацій

Останньою складовою маркетингової програми є розроблення концепції мркетингових комунікацій, що спирається на попередньо обрану основу для позицінування, визначену специфіку поведінки клієнтів (Таблиця 5.22)

Таблиця 5.22 - Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Цільові клієнти з великою ймовірністю вже користуються певними подібними сервісами.	Рекомендації знайомих, прямий зв'язок з цільовими клієнтами.	Позиціювання на основі порівняння товару фірми з товарами конкурентів, Позиціювання на позитивних особливостях технології, Позиціювання за показниками якості	Привернути увагу споживача, ознайомити з усіма перевагами використання продукту, переконати в унікальності продукту.	Презентація переваг та співвідношення ціна-якість. Зменшуємо затрати на ребалансування, враховуємо всі необхідні умови та мінімізуємо к-ть незадоволених клієнтів.

Отже було сформовано ринкову (маркетингову) програму, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого буде впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

5.6 Висновки

В даному розділі проведено аналіз створення та виведення на ринок стартап проекту на основі програмного продукту, який було розроблено в рамках магістерської дисертації. В межах цього аналізу було розроблено опис самої ідеї проекту, визначено загальні напрями використання товару, проаналізовано ринкові можливості щодо впровадження проекту, визначено відмінності від конкурентів та розроблено стратегію виходу на ринок. Узагальнюючи проведений аналіз, можна зазначити, що є можливість ринкової комерціалізації проекту. Наявний попит, динаміка ринку зростає, а також висока рентабельність роботи на ринку. З огляду на потенційні групи клієнтів та високий рівень конкурентоспроможності проекту є достатні перспективи для впровадження стартапу. Отже, подальша імплементація проекту є доцільною.

ВИСНОВКИ

В рамках роботи було проведено вичерпний аналіз літератури та існуючих методів навчання з підкріпленням, спеціалізованих на довгих горизонтах планування.

Серед робіт присвячених робототехніці, було виявлено багато робіт, що намагаються вирішити цю проблему. Серед них було взято метод навчання із моделями на основі темпоральних різниць [19].

Було створено програмний продукт для проведення експериментів та аналізу їх результатів. Існуючий продукт інтегрований із бібліотекою примітивів для автоматичної торгівлі цінними паперами – TensorTrade. За допомогою глибокого навчання із підкріпленням та довгострокового планування, було отримано нейромережу, що наближує стратегію, яка досягає кращих результатів та метрик на вибраному наборі даних.

Перевагою запропонованого підходу є довший період прибутковості. В той час як, у [44] перетренування моделі відбувається кожні 5 років при розмірі навчальної вибірки у 10 років, та 1 рік при розмірі навчальної вибірки у 4 роки, запропонований метод виходить на плато за 2-3 роки при розмірі вибірки у 2 роки. Такий ефект досягається через використання Моделей із Темпоральними Різницями [19] в основі методу.

Це було встановлено шляхом проведення великої кількості експериментів для статистичної достовірності. В порівняння були включені інші методи глибокого навчання із підкріпленням, а також класичні методи автоматичної торгівлі цінними паперами.

Був проведений детальний аналіз отриманих моделей та їх ефективності в рамках поставленої задачі. Крім того, після аналізу було встановлено перелік напрямків для покращення існуючого результату.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. A. Krizhevsky, I. Sutskever, G. Hinton Imagenet classification with deep convolutional neural networks. Nevada, Lake Tahoe: *Advances in neural information processing systems*, 2012. P. 1097-1105.
2. K. Simonyan, A. Zisserman Very deep convolutional networks for large-scale image recognition (arXiv preprint arXiv:1409.1556), 2014.
3. K. He, X. Zhang, S. Ren, J. Sun Deep residual learning for image recognition. Nevada, Las Vegas: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. P. 770-778.
4. S. Ren, K. He, R. Girshick, J. Sun Faster r-cnn: Towards real-time object detection with region proposal networks. Canada, Montreal: *Advances in neural information processing systems*, 2015. P. 91-99.
5. K. He, G. Gkioxari, P. Dollár, R. Girshick Mask r-cnn. Hawaii, Honolulu: *Proceedings of the IEEE international conference on computer vision*, 2017. P. 2961-2969.
6. L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. Yuille Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 2017. P. 834-848.
7. O. Ronneberger, P. Fischer, T. Brox U-net: Convolutional networks for biomedical image segmentation. Germany, Munich: *International Conference on Medical image computing and computer-assisted intervention*, 2015. P. 234-241.
8. G. Cybenko Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* 1989, Vol 2. No. 4. P. 303-314.
9. G. Tesauro Temporal difference learning and TD-Gammon. *Communications of the ACM* 1995, Vol 6. No. 2. P. 58-68
10. D. Silver, A. Huang, C. Maddison Mastering the game of Go with deep neural networks and tree search. *Nature.* 2014. Vol. 7589. No. 529. P. 484.

11. D. Silver, T. Hubert, J. Schrittwieser A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science.* 2018. Vol. 362, No. 6419. P. 1140-1144.
12. N. Savinov, A. Dosovitskiy, V. Koltun Semi-parametric topological memory for navigation. (arXiv preprint arXiv:1803.00653), 2018.
13. Japanese Nursing Association. Report on nursing in Japan, 2016. Дата оновлення 03.09.2016. URL: https://www.nurse.or.jp/jna/english/pdf/nursing-in-japan2016.pdf (дата звернення: 09.05.2019).
14. P. Anderson, A. Chang, D. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka On evaluation of embodied navigation agents. (arXiv preprint arXiv:1807.06757), 2018.
15. E. Feinberg, P. Kasyanov, M. Zgurovsky Convergence of value iterations for total-cost mdps and pomdps with general state and action sets. *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning* (ADPRL), 2014. P. 1-8.
16. V. Mnih, A. Badia, M. Mirza Asynchronous methods for deep reinforcement learning. New York: *International conference on machine learning*, 2016. P. 1928-1937.
17. J. Schulman, F. Wolski, P. Dhariwal Proximal policy optimization algorithms. (arXiv preprint arXiv:1707.06347), 2017.
18. J. Schulman, S. Levine, P. Abbeel Trust region policy optimization. France, Lille: *International Conference on Machine Learning*, 2015. P. 1889-1897.
19. V. Pong, S. Gu, S., M. Dalal, S. Levine. Temporal difference models: Model-free deep rl for model-based control. *arXiv preprint arXiv:1802.09081*, 2018.
20. S. Fujimoto, H. van Hoof, D. Meger Addressing function approximation error in actor-critic methods. (arXiv preprint arXiv:1802.09477), 2018.
21. T. Haarnoja, A. Zhou, P. Abbeel Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor (arXiv preprint arXiv:1801.01290), 2018.
22. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller Playing atari with deep reinforcement learning. (arXiv preprint arXiv:1312.5602), 2013.

23. M. Bellemare, W. Dabney, R. Munos A distributional perspective on reinforcement learning. *Proceedings of the 34th International Conference on Machine Learning* 2017, Vol. 70 No. 34 P. 449-458.
24. W. Dabney, M. Rowland, M. Bellemare Distributional reinforcement learning with quantile regression. California, San Diego: *Second AAAI Conference on Artificial Intelligence*, 2018. P. 3095-3109.
25. M. Andrychowicz, F. Wolski, A. Ray Hindsight experience replay. California, Long Beach: *In Advances in Neural Information Processing Systems,* 2017. Vol.4. No 369. P. 5048-5058.
26. S. Racanière, T. Weber, D. Reichert Imagination-augmented agents for deep reinforcement learning. California, Long Beach: *Advances in neural information processing systems*, 2017. P. 5690-5701.
27. A. Nagabandi, G. Kahn, R. Fearing Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. P. 7559-7566.
28. S. Ross, G. Gordon, D. Bagnell A reduction of imitation learning and structured prediction to no-regret online learning. *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011. P. 627-635.
29. R. Williams Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 1992. Vol. 8. No. 3-4. P. 229-256.
30. J. Baxter, P. Bartlett Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research* 15, 2001. P. 319-350.
31. Z. Kolter, A. Ng Near-Bayesian exploration in polynomial time. Canada, Montreal: *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009. P. 513-520.
32. A. Strehl, M. Littman An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences* 2008. Vol. 74. No. 8. P. 1309-1331.

33. D. Pathak, P. Agrawal, A. Efros Curiosity-driven exploration by self-supervised prediction. Hawaii, Honolulu: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017. P. 16-17.
34. B. Zhou, A. Lapedriza, A. Khosla Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence* 2017. Vol. 40, No. 6. P. 1452-1464.
35. E. Feinberg, P. Kasyanov, M. Zgurovsky Partially observable total-cost Markov decision processes with weakly continuous transition probabilities. *Mathematics of Operations Research* 2016. Vol. 41. No. 2. P. 656-681.
36. E. Feinberg, P. Kasyanov, N. Zadoianchuk Average cost Markov decision processes with weakly continuous transition probabilities. *Mathematics of Operations Research* 2012. Vol. 37. No. 4. P. 591-607.
37. O. Viskov, A. Shiryaev On controls which reduce to optimal stationary regimes. *Trudy Matematicheskogo Instituta imeni VA Steklova*, 1964. Vol. 71. No 8. PP. 35-45.
38. D. Blackwell Discrete dynamic programming. *The Annals of Mathematical Statistics*, 1962. Vol 33. No. 2. P. 719-726.
39. T. Lillicrap, J. Hunt, A. Pritzel Continuous control with deep reinforcement learning. (arXiv preprint arXiv:1509.02971), 2015.
40. R. Sutton and A. Barto. Reinforcement learning: An introduction. Cambrige, MA: MIT press, 2018. 338 p.
41. V. Konda and J. Tsitsiklis. "Actor-critic algorithms." In *Advances in neural information processing systems*, 2000, P. 1008-1014.
42. S. Fujimoto, H. Van Hoof, and D. Meger. "Addressing function approximation error in actor-critic methods." *arXiv preprint arXiv:1802.09477* (2018).
43. H. Yang, X. Liu, S. Zhong, and A. Walid. Deep reinforcement learning for automated stock trading: An ensemble strategy. *ACM International Conference on AI in Finance (ICAIF)*, 2020. P. 1036-1058.
44. Z. Zhang, S. Zohren, S. Roberts Deep reinforcement learning for trading. *The Journal of Financial Data Science*. 2020. Vol. 2. No. 2. P 25-40.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
from dataclasses import dataclass, field
from pathlib import Path

import av
import gym
from gym.spaces import Discrete, Box
import numpy as np
import torch
import torch.nn as nn

from ppo.rollout import TorchStorage
from ppo.statistics import Statistics, Timer


def ppo_update(
  params,
  model=None,
  opt=None
):
  assert not (model is None and opt is None), \
    "Either model or opt has to be specified"
  if opt is None:
    opt = torch.optim.Adam(
      model.parameters, lr=params.learning_rate, eps=params.eps)

  def _update_fn(model, storage, statistics):

    adv = storage.get_normalized_advantage(
      params.use_gae,
      params.discount_factor,
      params.gae_lambda
    )

    mean_value_loss  = 0
    mean_policy_loss = 0
    mean_entropy     = 0
    mean_loss        = 0
    mean_clip_frac   = 0
    mean_approx_kl   = 0
    mean_action_diff = 0

    batch_generator = storage.generate_fwd

    for epoch in range(params.ppo_epoch_num):
      for batch in batch_generator(adv, params.batch_size):
        (
          obs_prev,
          h_states,
          actions,
          value_preds_saved,
          returns,
          not_dones,
          log_pi_saved,
          adv_target,
        ) = batch

        policy_out, value_pred = model.policy_and_value(obs_prev)
        act, log_pi = model.select_action(policy_out, action=actions)
        entropy = model.entropy(policy_out).mean()

        r = (log_pi.unsqueeze(-1) - log_pi_saved).exp()
        r1 = adv_target * r
        r2 = adv_target * r.clamp(1.0 - params.clip_ratio,
                                  1.0 + params.clip_ratio)
```

```
                policy_loss = -torch.min(r1, r2).mean()

                if params.clip_value_loss:
                    value_pred_clipped = (
                        value_preds_saved +
                        torch.clamp(
                            value_pred - value_preds_saved,
                            -params.clip_value_loss,
                            +params.clip_value_loss
                        )
                    )
                    v_loss = torch.pow(value_pred - returns, 2)
                    v_loss_clip = torch.pow(value_pred_clipped - returns, 2)
                    value_loss = torch.max(v_loss, v_loss_clip).mean() * 0.5
                else:
                    value_loss = torch.pow(
                        value_pred - returns, 2).mean() * 0.5

                loss = value_loss * params.value_loss_coef + policy_loss
                reg = entropy * params.entropy_coef
                with torch.no_grad():
                    clip_frac = torch.gt(
                        (r - 1.0).abs(), params.clip_ratio).float().mean()
                    approx_kl = (
                        log_pi.unsqueeze(-1) - log_pi_saved).pow(2).mean() * 0.5
                    action_diff = (act - actions).pow(2).mean()

                opt.zero_grad()
                (loss - reg).backward()
                nn.utils.clip_grad_norm_(model.parameters(), params.clip_grad_norm)
                opt.step()

                mean_value_loss += value_loss.item()
                mean_policy_loss += policy_loss.item()
                mean_entropy += entropy.item()
                mean_loss += loss.item()
                mean_clip_frac += clip_frac.item()
                mean_approx_kl += approx_kl.item()
                mean_action_diff += action_diff.item()

            n_loop_iter = (
                params.ppo_epoch_num *
                params.traj_len *
                params.n_workers //
                params.batch_size
            )

            mean_value_loss /= n_loop_iter
            mean_policy_loss /= n_loop_iter
            mean_entropy /= n_loop_iter
            mean_loss /= n_loop_iter
            mean_clip_frac /= n_loop_iter
            mean_approx_kl /= n_loop_iter
            mean_action_diff /= n_loop_iter

            mean_log_pi = storage._st["log_pi"].mean().item()
            mean_v_pred = storage._st["value_pred"].mean().item()
            mean_ret = storage._st["returns"].mean().item()
            mean_adv = adv.mean().item()

            statistics.add_agent_data({
```

```
                "loss":       mean_loss,
                "value_loss":  mean_value_loss,
                "policy_loss": mean_policy_loss,
                "entropy":     mean_entropy,
                "clip_frac":   mean_clip_frac,
                "approx_kl":   mean_approx_kl,
                "log_pi":      mean_log_pi,
                "value_pred":  mean_v_pred,
                "returns":     mean_ret,
                "advantages":  mean_adv,
                "logstd":
model.logvar.mean().item(),
                "act_diff":    mean_action_diff,
              })

        return _update_fn



      def save_video(frames, name, fps=15):
        container          =          av.open(str(name),
mode="w")
        stream = container.add_stream("mpeg4",
rate=fps)
        shape = frames[0].shape
        dtype = frames[0].dtype
        if len(shape) == 2:
          fmt = "gray"
        elif len(shape) == 3:
          fmt = "rgb24"
        else:
          raise ValueError("Frame may have 2 or
3 "
                   f"dimensions,                not
{len(shape)}")

        if dtype != "uint8":
          warnings.warn(f"Frame dtype has to be
uint8, not {dtype}. "
                  "It will be min-max normalized
and casted.")
          def norm_uint8(x):
            x = x - x.min()
            x /= x.max()
            x *= 255
            return x.astype(np.uint8)

          norm_fn = norm_uint8
        else:
          norm_fn = lambda x: x

        stream.width = shape[1]
        stream.height = shape[0]
        stream.pix_fmt = "yuv420p"
        for frame in frames:
          frame                                =
av.VideoFrame.from_ndarray(norm_fn(frame),
format=fmt)
          for packet in stream.encode(frame):
            container.mux(packet)
        for packet in stream.encode():
          container.mux(packet)
        container.close()



    @dataclass
    class Params_PPO:
      n_workers:      int   = 8
      traj_len:       int   = 256
      use_gae:        bool  = True
      discount_factor: float = .99
      gae_lambda:     float = .95
      ppo_epoch_num:  int   = 4
      batch_size:     int   = 15
      clip_ratio:     float = .2
      clip_value_loss: float = .2
      clip_grad_norm:  float = .5
      entropy_coef:    float = .01
      value_loss_coef: float = .5
      learning_rate:   float = 7e-4
      eps:            float = 1e-5
```

```
@dataclass
class Params_Stats:
  run_path:  Path = field(default=Path())
  ckpt_path: Path = field(default=Path())

  freq_dump: int = 10
  freq_vid:  int = 250
  freq_ckpt: int = 50


@dataclass
class Params:
  ppo:                Params_PPO                =
field(default_factory=Params_PPO)
  stats:              Params_Stats              =
field(default_factory=Params_Stats)



def ppo_train(
  params,
  model,
  env,
  max_episodes=None,
  max_updates=None,
  max_steps=None
):

  assert not (max_episodes is None and
        max_updates is None and
        max_steps is None), \
    "Either  max_episodes  or  max_updates
has to be specified"
  max_episodes    =    max_episodes    or
float("inf")
  max_updates     =     max_updates     or
float("inf")
  max_steps = max_steps or float("inf")

  storage = TorchStorage(
    params.ppo.traj_len,
    params.ppo.n_workers,
    env.observation_space.shape,
    env.action_space,
    1
  )

  statistics = Statistics(
    params.ppo.n_workers,

tb_run_experiment=params.stats.run_path / "tb",
  )

  update_fn = ppo_update(
    params.ppo,
    model=model
  )

  obs_cur = env.reset_all()
  done_cur     =     [False     for     _     in
range(params.ppo.n_workers)]
  n_updates = 0

  best_metric = -float("inf")

  total_timer = Timer()
  perf_timer = Timer()

  total_timer.start()
  while (env.n_episodes < max_episodes
     and n_updates < max_updates
     and statistics.total_steps < max_steps):
    # TODO: rnn state passing to network
    with torch.no_grad():
      policy_out,          value_pred          =
model.policy_and_value(
        torch.FloatTensor(obs_cur))
      act,          log_pi_chosen          =
model.select_action(policy_out)
    obs_prev = obs_cur
    done_prev = done_cur
```

```
            ### TODO: is_good mask (bad_transition)
            # Rollout layout:
            # o0 -> o1 -> o2 -> o3 -> o4 -| o0
            # o0 -> o1 -> o2 -> o3 -| o0 -> o1
            # o0 -> o1 -> o2 -> o3 -> o4 -| o0
            # o0 -> o1 -| o0 -> o1 -> o2 -> o3

            if False and isinstance(env.action_space, Box):
              act = np.clip(act.cpu().numpy(),
                      env.action_space.low,
                      env.action_space.high)
            else:
              act = act.cpu().numpy()

            perf_timer.start()
            obs_cur, reward, done_cur, info = env.step_nonstop(act)
            env_time = perf_timer.get_elapsed_seconds()

            statistics.add_env_data({
              "time_per_env": env_time,
              "actions":    act,
              "reward":     reward,
              "done":       done_prev,
              "info":       info,
              "obs":        obs_prev,
            })


            if isinstance(env.action_space, Discrete):
              act = torch.LongTensor(act)
              act.unsqueeze_(-1)
            else:
              act = torch.FloatTensor(act)

            storage.save_rollout({
                "obs": torch.FloatTensor(obs_prev),
                "value_pred": torch.FloatTensor(value_pred),
                "log_pi": torch.FloatTensor(log_pi_chosen).unsqueeze(-1),
                "rewards": torch.FloatTensor(reward).unsqueeze(-1),
                "actions":   act,
                "dones": torch.FloatTensor(done_prev).unsqueeze(-1),
              })

            if storage.i_step == storage.traj_len - 1:

              with torch.no_grad():
                value_pred = model.value(torch.FloatTensor(obs_cur))
              storage.finalize_storage(
                value_pred.view(-1, 1),
                torch.FloatTensor(done_cur).view(-1, 1),
              )
              update_fn(model, storage, statistics)
              n_updates += 1

              if n_updates % params.stats.freq_dump == 0:
                statistics.reduce(total_timer.get_elapsed_seconds())
                statistics.dump()

              if n_updates % params.stats.freq_vid == 0:
                env.start_recording(1, size=300, mode="rgb_array")

              if n_updates % params.stats.freq_ckpt == 0:
                if statistics.stats["reward/epi_mean"] > best_metric:
```

```
model.save(params.stats.ckpt_path / "best.pth")
                    model.save(params.stats.ckpt_path
/ f"{n_updates}.pth")


                res = env.try_get_recordings()
                if res:
                  statistics.dump_video(res, "episodes",
fps=15, size=None)
        #save_video(res[0],   params.stats.run_path   /
f"{n_updates}.mp4")
```